



HAL
open science

A Hybrid Improved Manta Ray Foraging Optimization With Tabu Search Algorithm for Solving the UAV Placement Problem in Smart Cities

Amylia Ait Saadi, Assia Soukane, Yassine Meraihi, Asma Benmessaoud
Gabis, Amar Ramdane-Cherif

► **To cite this version:**

Amylia Ait Saadi, Assia Soukane, Yassine Meraihi, Asma Benmessaoud Gabis, Amar Ramdane-Cherif. A Hybrid Improved Manta Ray Foraging Optimization With Tabu Search Algorithm for Solving the UAV Placement Problem in Smart Cities. *IEEE Access*, 2023, 11, pp.24315-24342. 10.1109/ACCESS.2023.3255793 . hal-04059709

HAL Id: hal-04059709

<https://hal.science/hal-04059709>

Submitted on 5 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Received 12 February 2023, accepted 2 March 2023, date of publication 10 March 2023, date of current version 15 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3255793

RESEARCH ARTICLE

A Hybrid Improved Manta Ray Foraging Optimization With Tabu Search Algorithm for Solving the UAV Placement Problem in Smart Cities

AMYLIA AIT SAADI^{1,2}, ASSIA SOUKANE³, YASSINE MERAIHI¹,
ASMA BENMESSAOUD GABIS⁴, AND AMAR RAMDANE-CHERIF²

¹LIST Laboratory, University of M'Hamed Bougara Boumerdes, Boumerdes 35000, Algeria

²LISV Laboratory, University of Paris-Saclay, 78140 Velizy, France

³ECE Paris School of Engineering, 75015 Paris, France

⁴Laboratoire des Méthodes de Conception des Systèmes, Oued Smar, Algiers 16309, Algeria

Corresponding author: Amylia Ait Saadi (a.aitsaadi@univ-boumerdes.dz)

This work was supported in part by the Directorate General for Scientific Research and Technological Development (DG-RSDT) of Algeria, and in part by the "Action Doctorale Internationale (ADI) 2021" project funded by the Initiative D'EXcellence (IDEX) Paris-Saclay under Grant ANR-11-IDEX-0003-02.

ABSTRACT The concept of smart cities is to enhance the life quality of residents and provide efficient services by integrating advanced information and communication technologies, autonomous robots, Internet of Things (IoT) devices, etc. Unmanned Aerial Vehicles (UAVs) are a class of autonomous flying mobile robots that bring a lot of benefits to smart cities due to their mobility, accessibility, autonomy, and many other advantages. Their integration allows for accomplishing hard and complex tasks that humans or other entities are not able to complete. In most applications, multiple connected UAVs are required to build a network under which missions are completed and tasks are shared. This network can be established by finding the optimal UAV placement that meets some requirements such as user coverage, UAV connectivity, energy, and load distribution. In this paper, we present a hybrid algorithm, called IMRFO-TS, based on the hybridization of Improved Manta Ray Foraging Optimization (IMRFO) with the Tabu Search (TS) algorithm for solving the UAV placement problem in a smart city. First, the tangential control strategy is incorporated into the original MRFO algorithm to enhance its convergence speed and explore the search space effectively. Second, The TS algorithm is hybridized with the IMRFO algorithm to increase the exploitation capability of IMRFO and improve the best solution (placement quality) obtained after each iteration. The performance of the proposed IMRFO-TS algorithm is validated using 52 benchmarks considering the fitness value, coverage, connectivity, energy consumption, and load distribution parameters. Compared to eight well-known optimization meta-heuristics such as the original MRFO, TS algorithm, Bat Algorithm (BA), Firefly Algorithm (FA), Grey Wolf Optimization (GWO), Sine Cosine Algorithm (SCA), Whale Optimization Algorithm (WOA), and Reptile Search Algorithm (RSA), the results of the experiments revealed the significant superiority of the proposed IMRFO-TS algorithm by obtaining promising solutions (optimal positions of UAVs) in the majority of the cases.

INDEX TERMS Manta ray foraging optimization, Tabu search, unmanned aerial vehicles (UAVs), UAV placement.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenzhou Tang.

I. INTRODUCTION

Nowadays, the subject of smart cities has become increasingly interesting, whether from research or industrial perspectives. In fact, the population today is continuously growing, and people tend to move from rural regions to big cities, whether for social, economic, or other motivations. As a result of this phenomenon, managing and handling services in cities human beings become more difficult and complex. This issue requires deploying intelligent and centralized system management to handle it. The use of advanced technologies in cities gives rise to the concept of smart cities. By definition, the smart city is a developed urban area where several technologies are integrated to enhance residents' lifestyles and service quality. The improvements in smart cities are growing for two simple reasons. First, manufacturers in collaboration with researchers, focus more on designing and proposing efficient solutions with reduced costs for smart cities. Second, humans continuously trust devices, machines, and robots for operations and applications, especially autonomous ones like drones.

Drones or Unmanned Aerial Vehicles (UAVs) are considered important components of smart cities where they have been successfully integrated. The kinematic, autonomy, mobility, and flexibility of UAVs facilitate their use in various applications such as data collection, monitoring, disaster, emergency, delivery services, and many other services. Most of these services require several UAVs for efficient use and information exchange either between them or with the service provider (infrastructure) or users (clients). Therefore, multiple UAVs cooperate and form a network called Flight Ad-Hoc Network (FANET) to accomplish the task. The network is called Aerial Mesh Network (AMN) if the connection is established between all UAVs. Furthermore, UAVs can be equipped with 5G systems or IEEE 802.11 modules for establishing and maintaining communication in the AMN. Designing the AMN is challenging and complex because it changes according to several factors, such as the area where UAVs are used, the target coverage, the connectivity, UAV resources, and energy consumption. All these factors are related to the positioning of UAVs in the first place. Increasing UAVs heights increases the coverage area but decreases the source power of UAVs. Low altitude level preserves both energy and UAVs' safety but decreases the coverage quality. Users are connected to the closest UAV that covers the area. According to user distribution, users are attached to some UAVs more or less than others. In this case, the load of UAVs is not equal. Some of them consume more energy and resources for communications until the resources will be insufficient. At this point, the overloaded UAV will not be able to operate and quit the network impacting the connectivity quality. The connectivity parameter depends on the number of deployed UAVs and their positions. Since it is established by a complete connection between them and based on the 3D distance that separates them. Consequently, the key to designing an optimal AMN is to find the optimal UAV placement that optimizes the mentioned objectives.

The problem of UAV placement is an NP-hard problem [1] where meta-heuristic algorithms can be applied to solve it and provide optimal solutions. Besides providing optimal solutions, meta-heuristics deal better with complex problems such as wireless networks where a complete mathematical model is hard to establish. Moreover, meta-heuristics perform in a reasonable time and are easy to implement. Thus, in this paper, we propose a meta-heuristic-based algorithm for solving the UAV placement problem.

Among existing meta-heuristics, the Manta Ray Foraging optimization algorithm is a recent Swarm Intelligence optimization algorithm developed by Zhao et al. [2] in 2020. Due to its simplicity and easy implementation, it was widely applied for solving optimization problems such as electrical engineering [3], [4], image processing [5], [6], mathematics [7], geology [8], feature selection [9], system identification [10], energy [11], [12], [13], networking [14], PID control [15], and many others. Similar to other meta-heuristics, the MRFO algorithm suffers from premature and slow convergence and attempts to fall to local optima. Moreover, the MRFO algorithm integrates the exploitation operators into the exploring search in the earlier stage of processing. This feature decreases the exploration ability and leads to local optima stagnation. To address this drawback, various solutions were proposed in the literature for various applications. In [12] and [16], MRFO was combined with Particle Swarm Optimization algorithm. Simulated Annealing was used as a local search to enhance the exploitation of MRFO in [17] and [18]. Both neural networks and deep neural networks were incorporated into the original MRFO, respectively in [19] and [20]. In [21], MRFO was hybridized with Salp Swarm Algorithm. A hybrid Harris Hawk Optimization (HHO) with MRFO algorithms was proposed in [22].

This study proposes a novel approach for solving the UAV placement problem based on Manta Ray Foraging Optimization. The proposed algorithm brings two significant improvements over the original MRFO algorithm. The MRFO algorithm is first adjusted with a non-linear control strategy adjustment to enhance exploration and exploitation balance. This strategy aims to improve the exploration of new areas of global optimization. To benefit from the strong exploitation of the Tabu search (TS) algorithm, we integrated it to improve MRFO. TS algorithm searches locally for a better solution where the initial solution has already been found. To our knowledge, there is no work hybridizing MRFO with TS algorithm in the literature and it is the first time the MRFO algorithm has been used to solve the UAV placement problem.

The main contribution of this paper is presented below:

- Defining a weighted fitness function that considers the user coverage, UAV connectivity, energy consumption, and load distribution metrics for solving the UAV placement problem as a multi-objective problem.
- Proposing a new hybrid algorithm, called Improved Manta ray Foraging Optimization Tabu Search

(IMRFO-TS) algorithm for determining the optimal placement of UAVs in smart city.

- Implementing the proposed IMRFO-TS and nine other meta-heuristics such as MRFO, Tabu Search (TS), Bat Algorithm (BA), Firefly Algorithm (FA), Grey Wolf Optimization (GWO), Sine Cosine Algorithm (SCA), Whale Optimization Algorithm (WOA), and Reptile Search Algorithm (RSA).
- Evaluating the efficiency of the proposed IMRFO-TS and the state-of-the-art meta-heuristics in 52 benchmarks regarding the fitness value, coverage, connectivity, energy, and load distribution metrics.
- The proposed IMRFO-TS outperforms the traditional optimization algorithms and significantly improves the deployment of UAVs network in 3D space.

The remainder of this paper is organized as follows. In Section II, a brief survey of the existing research related to the present study is presented. Section III elaborates the system model used in this work and details the mathematical model used to solve the UAV placement problem. In section IV, a description of the Mantra ray Foraging algorithm, Tabu search algorithm, and control parameter strategy is given. Section V focuses on the proposed Hybrid Mantra ray Foraging with Tabu search algorithms for solving the UAV placement problem. Section VI reports the simulation results and comparisons followed by conclusions and future works in Section VII.

II. LITERATURE REVIEW

The current trend of technology research is to enhance the quality and services in smart cities by deploying the UAVs network as an access layer. The main purpose of this deployment is to determine the optimal placement of UAVs for maximum efficiency. Since the UAV placement is an NP-hard problem [1], various meta-heuristics for solving this problem have been proposed in the literature. We will focus in this section on some previous works (see Table 1) proposed regarding this topic based on meta-heuristics.

Wang et al. [23] proposed an improved Tabu Search algorithm based on the integration of the complementary concept into the TS algorithm for solving the UAV placement issue. The proposed algorithm was evaluated using a different number of UAVs and end nodes. Simulation results demonstrated the efficiency of the improved Tabu Search algorithm compared to Tabu Search and branch-and-cut optimizer approaches considering the solution quality and execution time.

Strumberger et al. [24] used the Elephant Herding Optimization (EHO) Algorithm to solve the UAV Placement problem. The EHO algorithm was assessed in four scenarios using a different number of drones and users. Simulated results show that EHO offers maximum coverage for users using a limited number of drones.

Authors in [25] applied the Tabu Search algorithm for solving the UAV placement problem. Tabu search Algorithm was evaluated using 4 UAVs and random users under different

connectivity requirements. Simulation results showed that the TS algorithm gives good results regarding the connectivity metric.

In the work of Reina et al. [26], authors proposed an ameliorated version of the GA algorithm, called Multi-Layout Multi-subpopulation Genetic Algorithm (MLMPGA), for solving the UAV placement issue. The efficiency of the MLMPGA algorithm was evaluated using different UAVs and users. The obtained results demonstrated that the MLMPGA algorithm outperforms GA, PSO, and HCA methods in terms of fitness value, coverage, fault tolerance, and redundancy.

The authors in [27] used both SA and GA for solving the UAV placement problem taking into account coverage and energy metrics. The performance of SA and GA are assessed using Java and Python, respectively, in an area of 80 kilometers squared with a different number of targeted users. Simulation results showed that the GA algorithm outperforms SA regarding fitness value and execution time.

In [28], A Social Spider Optimization (SSO) Algorithm was employed to solve the problem of UAV deployment. An evaluation of its effectiveness was conducted in three different areas serving a different number of users, and a comparison was made with the Random Search (RS) method and the Uniform Distribution application. SSO outperforms other meta-heuristics regarding fitness value, execution time, and covered users in simulations.

Yang et al. [29] applied the Differential Evolution (DE) algorithm to address the problem of the UAV placement. The performance of DE algorithm was validated in one case using 5 UAVs and 100 users. DE outperforms both Genetic Algorithm (GA) and PSO algorithms in terms of fitness value.

Gupta and Varma [30] used four meta-heuristics for optimizing the UAV placement in emergency cases, named Multi-objective Particle Swarm Optimization (MOPSO), Non-dominated Sorting Genetic Algorithm 2 (NSGA-II), Strength Pareto Evolutionary Algorithm 2 (SPEA2), and Pareto Envelope-based Selection Algorithm (PESA-II). The four algorithms were validated in three scenarios (small, medium, and large-scale) with different users. Test results demonstrated that, based on fitness values, SPEA2 is suitable for a small number of users, while NSGA-II is appropriate for a medium and large number of users.

In another work, the same authors [31] suggested two hybrid algorithms for an optimal UAV placement, called HWWO-HSA and HGA-SA. The HWWO-HSA scheme is based on the hybridization of Water Wave Optimization (WWO), Harmony Search (HS), and SA algorithms. The GA-SA algorithm is the result of the hybridization of GA with SA algorithms. Both proposed algorithms were tested in three scenarios using twelve different number of users. Results confirmed the superiority of the proposed algorithms in terms of fitness values compared to WWO, HS, SA, and GA methods.

Sawalmeh et al. [32] suggested GA-based and PSO-based clustering algorithms for solving the UAV placement issue. Both algorithms were assessed using a various number of

drones and ground users. Simulation results proved that the PSO-based clustering algorithm outperforms GA-based, ABC-based, and k-mean clustering algorithms in terms of computational complexity and cost.

Authors in [33] applied the Particle Swarm Optimization (PSO) algorithm for solving the UAVs-based Quality of Service (QoS) placement problem. The PSO algorithm was evaluated in four different areas with multiple users. Simulation results showed that the PSO algorithm provides good results in finding optimal positions of UAVs regarding the coverage and the requirements of QoS.

In [34], two optimization approaches in a Continuous-data range (OFSAC-PSO and OFSAC-EML) and two approaches in a Discrete-data range (OFSAD-PSO and OFSAD-EML) are developed for optimizing the UAV placement problem. Both approaches are based on Particle Swarm Optimization (PSO) and ElectroMagnetism-Like (EML) algorithms. These algorithms were applied in one area using 2601, 676, and 7734 users. Results showed that the OFSAC-PSO algorithm outperforms WOA, OFSAC-EML, OFSAD-PSO, and OFSAD-EML based on the compared metrics.

Ouamri et al. [35] applied the GWO algorithm for solving the UAV placement issue. The effectiveness of the GWO algorithm was evaluated using 10 UAVs and 200 users. Simulation results showed the performance of the GWO algorithm by covering more than 85% of users.

In [36], authors proposed a multi-objective Cuckoo Search algorithm for optimal placement of UAVs. The proposed algorithm was evaluated using a different number of drones and ground users. The proposed algorithm gives competitive results regarding the coverage, energy, and network efficiency metrics.

Wei et al. [37] applied Deep Q-learning (DQN) approach for solving the UAV, Unmanned Surface Vehicle (USV), and Unmanned Underwater Vehicle (UUV) placement issue. The efficiency of DQN algorithm was assessed in one scenario using 1 UAV, 1 USV, and 3 UUVs. Compared to ACO, Double DQN, and Dueling DQN algorithms, DQN approach outperforms in terms of success rate, training time, and energy optimization.

Liu et al. [38] proposed a Proximal Stochastic Gradient Descent Based Alternating approach for solving the UAV placement problem. The proposed approach was tested in area where the users are grouped into 15 cells served by Base Stations. the proposed approach achieves good results in terms of fitness value and coverage rate.

The summary of these works is presented in Table 1

III. UAV PLACEMENT PROBLEM FORMULATION

The purpose of this section is to describe the UAV placement problem in more detail. To begin with, we will present the system model we used in this study. Afterward, we will mathematically define the objective function that includes user coverage, UAV connectivity, energy, and load distribution requirements.

A. UAV SYSTEM MODEL

We consider in an urban area a set of users $G = \{g_1, g_2, \dots, g_m\}$, where m is the total number of users and a set of UAVs $U = \{u_1, u_2, \dots, u_n\}$, where n is the total number of UAVs. Each UAV $u_j, j \in \{1, 2, \dots, n\}$ is located at the coordinate (x_j, y_j, z_j) within the boundaries of the served area $[L, W, H]$, where L, W , and H represent the length, width and height limits, respectively. Similarly, each user $g_i, i \in \{1, 2, \dots, m\}$ is located within the same area at the coordinate (x_i, y_i) .

In this system model, we assume that each UAV $u_j, j \in \{1, 2, \dots, n\}$, is equipped with an antenna for wireless communications characterized by a maximum transmission range R_{max} . The purpose of this communication is to form links with users to provide services and also to share information with other UAVs forming an Aerial Mesh Network (AMN) as shown in Figure 1.

B. OBJECTIVE FUNCTION

To solve the UAV placement problem, an objective function is designed to optimize simultaneously the four considered objectives which are user coverage, UAV connectivity, energy consumption, and equal load balancing. Their definitions are given below:

1) USER COVERAGE

UAVs are said to cover a user when they are located at a Euclidean distance d shorter than or equal to their coverage range in the horizontal plane (r) from it. Therefore, the coverage model considers a boolean variable for the connectivity between the UAV and the user. It is mathematically given by Equation (1).

$$c_{g_i, u_j} = \begin{cases} 1, & \text{if } d(g_i, u_i) \leq r_j \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

where $d(g_i, u_j)$ is the euclidean distance between the UAV u_j and the user g_i in the horizontal plane formulated in Equation (2). r_j stands for the UAV coverage range related to both altitude and visibility angle as shown in Figure 2. It can be calculated using Equation (3)

$$d(g_i, u_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2)$$

$$r_j \leq z_j \cdot \tan\left(\frac{\theta}{2}\right) \quad (3)$$

where z_j represents the altitude level of the UAV at the position (x_j, y_j) . θ is the visibility angle. The total user coverage in this model C_v is defined as follows:

$$C_v = \sum_{j=1}^n c_{g, u_j} \quad (4)$$

Our first objective is to find the (x_j, y_j, z_j) coordinate that maximizes C_v by:

$$\max C_v = \sum_{j=1}^n \max c_{g, u_j} \quad (5)$$

TABLE 1. Comparative table of some existing UAV Placement works.

Algorithms	Reference	Coverage	Energy	Connectivity	Load balancing
Improved TS	Wang et al. [23]	✗			
EHO	Strumberger et al. [24]	✗			
TS	Ur Rahman et al. [25]	✗	✗		
MLMPGA	Reina et al. [26]	✗			
SA and GA	Al-Turjman et al. [27]	✗	✗		
SSO	Chaalal et al. [28]	✗			
MOPSO, NSGA-II, SPEA2, and PESA-II	Gupta and Varma [30]	✗	✗		
DE	Yang et al. [29]	✗			✗
HWWO-HSA and HGA-SA	Gupta and Varma [31]	✗	✗		✗
PSO-C and GA-C	Sawalmeh et al. [32]	✗			
PSO	Mayor et al. [33]	✗			
OFSAC-PSO, OFSAC-EML, OFSAD-PSO, and OFSAD-EML	Recep Ozdag [34]	✗			
GWO	Ouamri et al. [35]	✗			
MOCS	Mahajan et al. [36]	✗	✗	✗	
DQN	Wei et al. [37]	✗	✗	✗	
ProxSGD-based alternating	Liu et al. [38]	✗	✗		
IMRFO-TS	Proposed method	✗	✗	✗	✗

TABLE 2. The main notations used in this paper.

Parameter	Description
UAV-BS	Unmanned Aerial Vehicle- Base Station
GU	Ground User
n	The total number of UAVs
m	The total number of users
c_{g_i, u_j}	The coverage cost of the i -th user with the j -th UAV
$d(g_i, u_j)$	The distance between the i -th user and the j -th UAV
r_j	The coverage range of the j -th UAV
z_j	The j -th UAV's height
θ	Visibility angle of UAV
Cv	The total coverage cost
ct_{u_j, u_k}	The connectivity cost of the j -th UAV with the k -th UAV
R_{max}	The maximum transmission range of the UAV in 3D plane
$d(u_j, u_k)$	The distance between the j -th and k -th UAVs
Cn	The total connectivity cost
E_j	Energy consumed of the UAV j during the flight
α	The minimum power required for the UAV
β	Motor speed multiplier
T	Flight time
P_{max}	Maximum motor power
s_j	The j -th UAV speed
En	The total energy consumed
ELD	Equal load distribution
g^{u_j}	The total covered user by the j -th UAV

2) UAV CONNECTIVITY

The nodes in a network are said to be connected when they have a path between them. A connected network has no unreachable nodes. The connectivity between nodes or users can be done by connecting the deployed UAVs.

The connectivity model considers also a Boolean decision expressed in Equation (6) based on the distance between UAVs.

$$ct_{u_j, u_k} = \begin{cases} 1, & \text{if } d(u_j, u_k) < 2R_{max} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

where $d(u_j, u_k)$ is the euclidean distance between the UAV u_j and u_k .

$$d(u_j, u_k) = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2} \quad (7)$$

The total connectivity of the UAV network Cn is given by Equation (8).

$$Cn = \bigcup_{j=1}^n ct_{u_j, u_k} \quad (8)$$

The second objective is to find the (x_j, y_j, z_j) coordinate that maximizes Cn by:

$$\max Cn = \bigcup_{j=1}^n \max ct_{u_j, u_k} \quad (9)$$

3) ENERGY CONSUMPTION

Energy is an important parameter to take into consideration for the efficient deployment of UAVs network. Till today, modeling the UAV's energy consumption is a challenging issue that researchers have focused on. According to the measurements and tests done in [39], the closest model to reality of the energy consumed by the UAV u_j during the flight is formulated as follows [40]:

$$E_j = (\alpha + \beta \cdot z_j)T + P_{Max} \left(\frac{z_j}{s} \right) \quad (10)$$

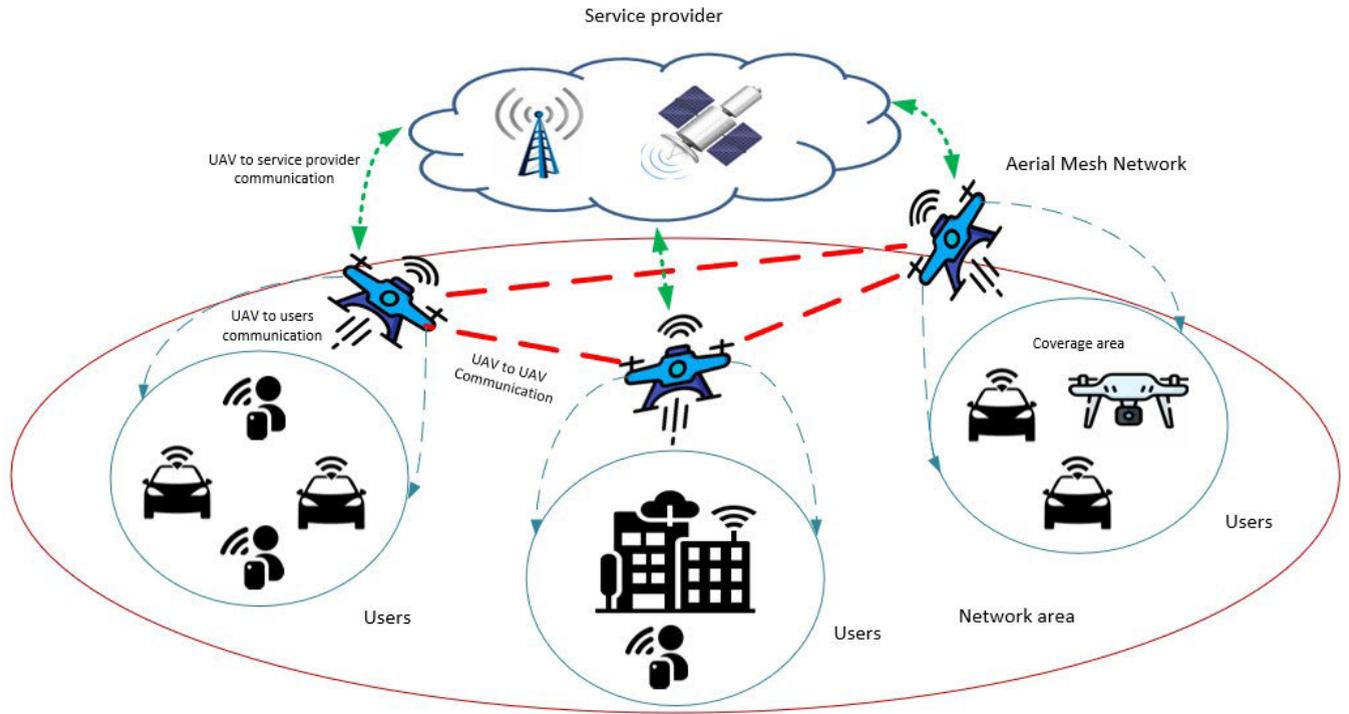


FIGURE 1. UAV-based wireless network architecture.

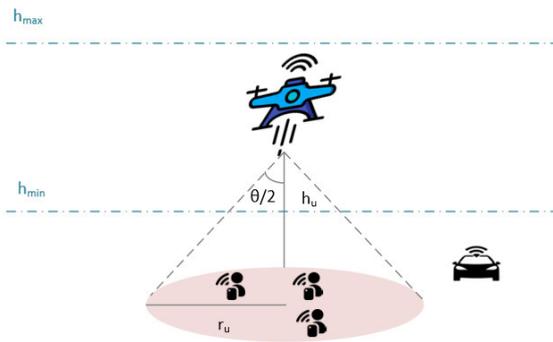


FIGURE 2. UAV Coverage.

where α is the minimum power required for the UAV to fly, β is the motor speed multiplier. T represents the flight time. P_{Max} denotes the maximum motor power. z_j and s_j are the height and the speed of the UAV u_j , respectively. The total energy consumed by the UAV network is given in Equation (11)

$$En = \sum_{j=1}^n E_j \tag{11}$$

The third objective is to determine the adequate UAV's height z_j to minimize the energy consumption by:

$$\min En \tag{12}$$

4) LOAD DISTRIBUTION

The physical characteristics of UAVs make them limited in terms of resources such as memory, wireless resources, etc. To ensure the optimal use of these resources and improve the homogeneity of the UAVs network, we introduced an equal load distribution to balance the exploitation of these resources. It is formulated mathematically in Equation (13).

$$ELD = \frac{1}{n} \sum_{j=1}^n (g^{u_j} - \frac{m}{n})^2 \tag{13}$$

where g^{u_j} is the total number of users covered by UAV u_j . The last objective is to minimize as much as possible the equal load distribution based on the UAV location by Equation (14).

$$\min ELD \tag{14}$$

Based on the four objectives defined, we formulated the objective function as a linear weighted function expressed in the following equation:

$$f(p_i) = \omega_1.Cv + \omega_2.Cn + \omega_3.E + \omega_4.ELD \tag{15}$$

where Cv represents the user coverage cost. Cn expresses the UAV connectivity cost. E and ELD represent energy and equal load distribution costs respectively. ω is a weight coefficient mathematically defined in Equation (16).

$$\omega = \sum_{l=1}^4 \omega_l = 1 \tag{16}$$

IV. PRELIMINARIES

This section describes the concept of Manta Ray Foraging Optimization Algorithm, Tabu Search algorithm, and Tangential Control adjustment strategy.

A. MANTA RAY FORAGING OPTIMIZATION ALGORITHM

Manta Ray Foraging Optimization Algorithm (MRFO) is a new nature-inspired meta-heuristic proposed in 2020 by Zhao et al. [2] which was applied for solving various optimization problems. The main inspiration of the MRFO algorithm comes from the behavior of manta rays in catching their prey. MRFO employs three main foraging strategies including chain foraging, cyclone foraging, and somersault foraging [2].

1) CHAIN FORAGING

In this strategy, Manta rays form a head-to-tail chain and start foraging by moving one after another. Except for the first individual, other manta rays move toward the food and the closest manta ray for cooperation. The mathematical expression of chain foraging is expressed as follows:

$$p_i^{t+1} = \begin{cases} p_i^t + r \cdot (p_{Best} - p_i^t) + \alpha \cdot (p_{Best} - p_i^t), & i = 1 \\ p_i^t + r \cdot (p_{i-1}^t - p_i^t) + \alpha \cdot (p_{Best} - p_i^t), & i = 2, \dots, N. \end{cases} \quad (17)$$

$$\alpha = 2 \cdot r \cdot \sqrt{\log(r)} \quad (18)$$

where p_i^t represents the i -th individual position at the iteration t . r is a random vector in the range $[0 - 1]$. α is a weight coefficient. p_{Best} represents the best position found so far.

2) CYCLONE FORAGING

This strategy is characterized by the spiral movement of Manta rays toward the food and the individual in front of it. It is mathematically represented in the following equation:

$$p_i^{t+1} = \begin{cases} p_i^t + r \cdot (p_{Best} - p_i^t) + \beta \cdot (p_{Best} - p_i^t), & i = 1 \\ p_i^t + r \cdot (p_{i-1}^t - p_i^t) + \beta \cdot (p_{Best} - p_i^t), & i = 2, \dots, N. \end{cases} \quad (19)$$

$$p_{rand} = Lb + r \cdot (Ub - Lb) \quad (20)$$

$$p_i^{t+1} = \begin{cases} p_i^t + r \cdot (p_{rand} - p_i^t) + \beta \cdot (p_{rand} - p_i^t), & i = 1 \\ p_i^t + r \cdot (p_{i-1}^t - p_i^t) + \beta \cdot (p_{rand} - p_i^t), & i = 2, \dots, N. \end{cases} \quad (21)$$

$$\beta = 2 \cdot e^{(r_1 \frac{T-t+1}{T})} \cdot \sin(2 \cdot \pi \cdot r_1) \quad (22)$$

where p_{rand} represents a random position in the space delimited by the lower and upper bounds Lb and Ub respectively. β donate a weight coefficient. r_1 is a random value within the range $[0 - 1]$. T is the maximum number of iterations.

3) SOMERSAULT FORAGING

In this case, the food position is shown as a pivot. The manta rays swim to and fro around the food and somersault to a new

position. This behavior is expressed as follows:

$$p_i^{t+1} = p_i^t + S \cdot (r_2 \cdot p_{Best} - r_3 \cdot p_i^t), \quad i = 1, \dots, N. \quad (23)$$

where S represents the somersault factor which is fixed to 2. r_2 and r_3 are random numbers in the range $[0 - 1]$.

Algorithm 1 The Pseudo-Code of Manta Ray Foraging Optimization Algorithm

```

1: Initialize MRFO parameters: Maximum number of iterations  $T$ , Population' size  $N$ , Dimension  $Dim$ ,  $\alpha$ ,  $\delta$ , etc.
2: Initialize the population of MRFO:  $X_i(i = 1, 2, \dots, N)$ 
3: Calculate the fitness value  $f(X_i)$ 
4: Determine the best position  $X_{best}$ 
5: while ( $t < T$ ) do
6:   for  $i = 1, 2, \dots, N$  do
7:     if  $rand < 0.5$  then
8:       if  $(\frac{t}{T}) < rand$  then
9:          $X_{rand} = Lb + rand \cdot (Ub - Lb)$ 
10:        Update the position  $X_i(t + 1)$  using Equation (21)
11:       else
12:         Update the position  $X_i(t + 1)$  using Equation (19)
13:       end if
14:     else
15:       Update the position  $X_i(t + 1)$  using Equation (17)
16:     end if
17:     Calculate the fitness value  $f(X_i(t + 1))$ 
18:     Update the best position  $X_{best}$ 
19:     Update the position  $X_i(t + 1)$  using Equation (23)
20:     Calculate the fitness value  $f(X_i(t + 1))$ 
21:     Update the best position  $X_{best}$ 
22:   end for
23:    $t = t + 1$ 
24: end while
25: return The best position  $X_{best}$ 

```

B. TABU SEARCH ALGORITHM

Tabu Search (TS) algorithm is a famous meta-heuristic proposed by Glover [41], widely applied for solving optimization problems [42], [43], [44], [45]. TS algorithm belongs to the single-based meta-heuristic category, which explores one candidate solution at each run [46]. Due to its characteristics, TS performs better as a local search algorithm by a neighborhood search approach. Starting from a candidate or initial solution, TS searches for potential other solutions nearby. Then, these solutions are evaluated one by one according to their fitness value. To make sure that TS algorithm doesn't explore solutions more than once, it creates a list to store the evaluated solutions. This list is updated if solutions don't meet evaluation criteria. In the end, the best solution which is not in the tabu list is selected.

The pseudo-code of the Tabu Search algorithm is presented in 2

Algorithm 2 The Pseudo-Code of Tabu Search Algorithm

- 1: Initialize TS parameters: Maximum number of iterations T , Neighbors' number n , Tabu List TL .
- 2: Generate initial solution X
- 3: Set the current solution X as the best solution X_{best}
- 4: **while** ($t < T$) **do**
- 5: **for** $i = 1, 2, \dots, n$ **do**
- 6: Generate a random neighbor X' of X
- 7: Evaluate neighbor' solution X'
- 8: Update TL
- 9: Update the best solution X_{best}
- 10: **end for**
- 11: $t = t + 1$
- 12: **end while**
- 13: **return** The best solution X_{best}

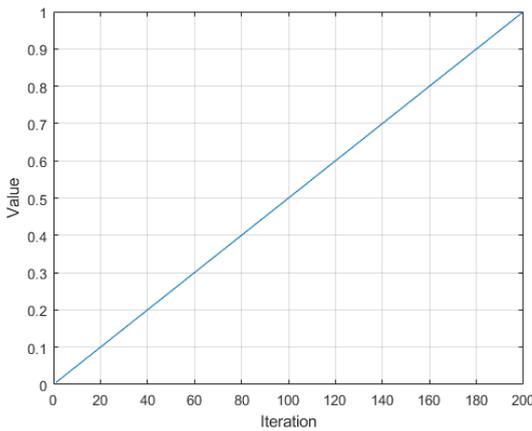


FIGURE 3. Linear control strategy.

C. NON-LINEAR CONTROL ADJUSTMENT STRATEGY

By definition, meta-heuristics are optimization algorithms that search for the optimal solution based on two strategies: exploration and exploitation search strategies. Exploration involves searching the unexplored areas of the feasible region, while exploitation involves searching the neighbourhoods of the promising region [47]. The performance of meta-heuristics depends mainly on their ability to balance both strategies. Mathematically, this balance is represented by a control parameter. In the MRFO algorithm, the term $(\frac{t}{T})$ defines the parameter that controls exploration and exploitation search. The variation of this parameter during the algorithm's execution is illustrated in Figure 3. To enhance the performance of the MRFO algorithm, we adopted a non-linear strategy using tangential variation as shown in Figure 4. As we can see from the figure, in the later stage of iterations, the control parameter (γ) is not close to 1. So, its value is not greater than *rand*. In this case, the algorithm remains in the exploration phase, searching for new and better solutions. Therefore, This improvement can enhance the performance of the MRFO algorithm by improving its exploration ability. Mathematically, it is represented by the

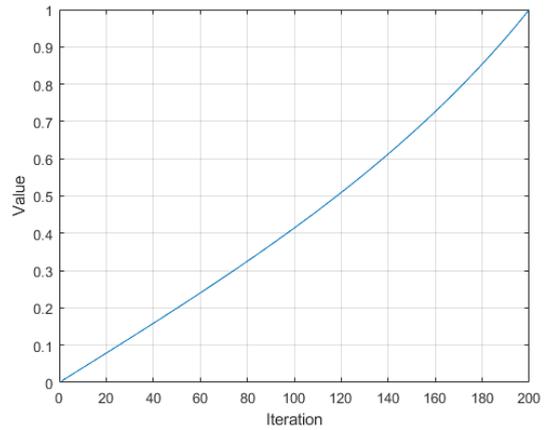


FIGURE 4. Tangential control strategy.

following equation:

$$\gamma = \tan\left(\frac{\pi}{4} \cdot \frac{t}{T}\right) \tag{24}$$

where t and T are the current and the maximum number of iterations, respectively.

V. HYBRID MANTA RAY FORAGING OPTIMIZATION ALGORITHM FOR UAV PLACEMENT

This section describes the implementing steps of our proposed algorithm (IMRFO-TS) for solving the UAV placement problem based on the hybridization of Improved MRFO and Tabu Search algorithms. To begin with, a non-linear control strategy is incorporated into the original MRFO for enhancing its exploration capability. Furthermore, to improve its search balance, the Tabu Search algorithm is combined as a local search algorithm with Improved MRFO for better exploitation. The pseudo-code of the proposed IMRFO-TS algorithm is given in Algorithm 3. Figure 5 illustrates the flow chart of IMRFO-TS algorithm.

The application of the IMRFO-TS algorithm for UAV placement involves 5 basic phases such as initialization, evaluation, update, local search application, termination, and solution representation.

A. INITIALIZATION

Similar to classical optimization functions, the important parameters should be given in a reasonable range for better efficiency starting with the search space. In UAV placement, the search space is set to be the 2D dimension of the area where UAVs are deployed, in addition to the height limitations of UAVs. In this space, each individual of the population represents a potential solution that is randomly initialized. The total population can be represented as follows:

$$Positions = \begin{bmatrix} P_{1,j} \\ P_{2,j} \\ \vdots \\ P_{N,j} \end{bmatrix} \tag{25}$$

where N is the total population size. $P_{i,j}$ denotes the i -th individual position that is represented in Equation (26)

$$P_{i,j} = \begin{cases} L_{min} \leq P_{i,j} \leq L_{max}, j = 1, \dots, n \\ W_{min} \leq P_{i,j} \leq W_{max}, j = n + 1, \dots, 2n \\ H_{min} \leq P_{i,j} \leq H_{max}, j = 2n + 1, \dots, 3n \end{cases} \quad (26)$$

where n stands for the number of UAVs. L_{min} and W_{min} are the minimum length and width of the search area. L_{max} and W_{max} donate the maximum length and width of the area. H_{min} stands for the minimum flight altitude that is fixed according to the application use and UAV safety. H_{max} represents the maximum flight altitude that is related directly to both visibility angle (θ) and maximum transmission range (R_{max}). It is mathematically expressed in Equation (27).

$$H_{max} = \frac{R_{max}}{\tan(\theta/2)} \quad (27)$$

B. EVALUATION

At this level, the population is initialized and the IMRFO-TS starts evaluating the individuals. For each individual, the fitness value is calculated using Equation (28).

$$Cost_i = f(P_i) \quad (28)$$

IMRFO-TS saves at this point the best individual found so far that corresponds to the best cost. The problem of UAV placement belongs to the category of maximization problems. Therefore, the best individual is defined as the individual where the maximum cost is reached. It is formulated in Equation (29).

$$P_{Best} = \arg \max f(P_i) \quad (29)$$

C. UPDATE

At the beginning of each iteration, the predefined control parameter ($\frac{t}{T}$) is replaced and updated using the proposed control strategy defined in Equation (24). This new control parameter strategy constitutes the first improvement for traditional MRFO to push better the exploration search strategy. According to the parameter values, individuals of the improved MRFO algorithm are updated in the same way as the original MRFO algorithm. The cost of each updated individual is calculated, and the best individual found is stored.

D. LOCAL SEARCH APPLICATION

To bring more balance to the improved MRFO algorithm, Tabu Search algorithm is added at the end of each iteration. Based on the previous best solution, Tabu Search makes its searches for better solutions in the neighborhood. If any better individual is found, the best individual is updated and the others are stored in its list.

E. TERMINATION

The proposed IMRFO-TS algorithm repeats the update and local search application phases until the maximum number of iterations is reached. Once the termination criteria is satisfied, IMRFO-TS displays the best solution found.

F. SOLUTION REPRESENTATION

This phase is an extra step that aims to present the solution in a format that facilitates the use of the solution for other purposes such as displaying and plotting. The best solution is given using the following representation:

$$P_{best} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \quad (30)$$

Algorithm 3 The Pseudo-Code of Hybrid Manta Ray Foraging Optimization Algorithm for UAV Placement

- 1: Initialize MRFO parameters: Maximum number of iterations T , Population' size N , Dimension Dim , α , δ , etc.
- 2: Initialize the population of MRFO: $P_i(i = 1, 2, \dots, N)$
- 3: Calculate the fitness value $f(P_i)$
- 4: Determine the best position P_{Best}
- 5: **while** ($t < T$) **do**
- 6: Update γ using Equation 24
- 7: **for** $i = 1, 2, \dots, N$ **do**
- 8: **if** $rand < 0.5$ **then**
- 9: **if** $\gamma < rand$ **then**
- 10: $X_{rand} = Lb + rand \cdot (Ub - Lb)$
- 11: Update the position $P_i(t + 1)$ using Equation (21)
- 12: **else**
- 13: Update the position $P_i(t + 1)$ using Equation (19)
- 14: **end if**
- 15: **else**
- 16: Update the position $P_i(t + 1)$ using Equation (17)
- 17: **end if**
- 18: Calculate the fitness value $f(P_i(t + 1))$
- 19: Update the best position P_{Best}
- 20: Update the position $P_i(t + 1)$ using Equation (23)
- 21: Calculate the fitness value $f(P_i(t + 1))$
- 22: Update the best position P_{Best}
- 23: **end for**
- 24: **Call TS** on P_{Best} as initial solution
- 25: Update the best solution P_{Best}
- 26: $t = t + 1$
- 27: **end while**
- 28: **return** The best position P_{Best}

G. TIME COMPLEXITY OF THE PROPOSED IMRFO-TS ALGORITHM

The time complexity of the IMRFO-TS depends on the problem dimension, the population size, and the maximum number of iterations. For every iteration, chain foraging, cyclone foraging, somersault foraging, and local search strategies are implemented. Therefore, the total time complexity of the IMRFO-TS algorithm is given as follows:

TABLE 3. Description of simulation parameter.

Simulation Parameter	Value
Area size	[1000, 1000]
Total number of users	[20, 60, 100, 140]
Total number of UAVs	[4 – 16]
UAVs transmission range	250m
Visibility angle	120°
E_0	100Kj
α	30
β	1
P_{Max}	85 Watt
s	2 m/s

$$O(IMRFO - TS) = O(T(O(chainforaging + cyclone foraging) + O(somersaultforaging) + O(localsearch)))$$

$$O(IMRFO - TS) = O(T(Nd + Nd) + T(nd))$$

$$O(IMRFO - TS) = O(Td(N + n))$$

where T stands for the maximum number of iterations. N and d represent the population size and the problem dimension, respectively. n donates the number of neighborhoods defined in TS algorithm.

VI. NUMERICAL EXPERIMENTS AND RESULTS

In this section, we introduce the simulation experiments, performance evaluation, and results of the proposed IMRFO-TS algorithm and other well-known meta-heuristic techniques, namely: TS [41], BA [48], FA, GWO [49], SCA [50], WOA [51], MRFO [2], and RSA [52]. These experiences aim to demonstrate the efficiency of the proposed IMRFO-TS algorithm by investigating how the IMRFO-TS adapts to the problem of UAV placement under various requirements. For this purpose, several scenarios have been considered using a different number of UAVs and users distributed randomly and non-uniformly in a square area of $1000 \times 1000 m^2$. The details of the simulation parameters are summarized in Table 3.

Before starting scenarios implementation, all optimization algorithms parameters are adapted and fixed according to Table 4. Afterward, simulations are running on MATLAB R2021b software with the computer processor Intel Core i7 2.90GHz, RAM 32 GB, and 64-bits Windows 11 operating system. The population size and the maximum number of iterations are set to 50 and 200, respectively.

We set different complexity ranges for the UAV placement problem. Starting with the simplest case using 4 UAVs and 20 users and ending with 140 users served by 16 UAVs. For each case, algorithms are evaluated according to their fitness value, coverage, connectivity, energy, and load distribution metrics. The evaluation was running 50 times and the results are presented in this section.

A. CASE OF 20 USERS

Table 5 reports the obtained results of 20 users covered by different number of UAVs. It can be observed that the proposed IMRFO-TS algorithm provides the best fitness value in most cases. The IMRFO-TS algorithm reaches its maximum when $U = \{4, 5\}$ which is considered as the appropriate use case

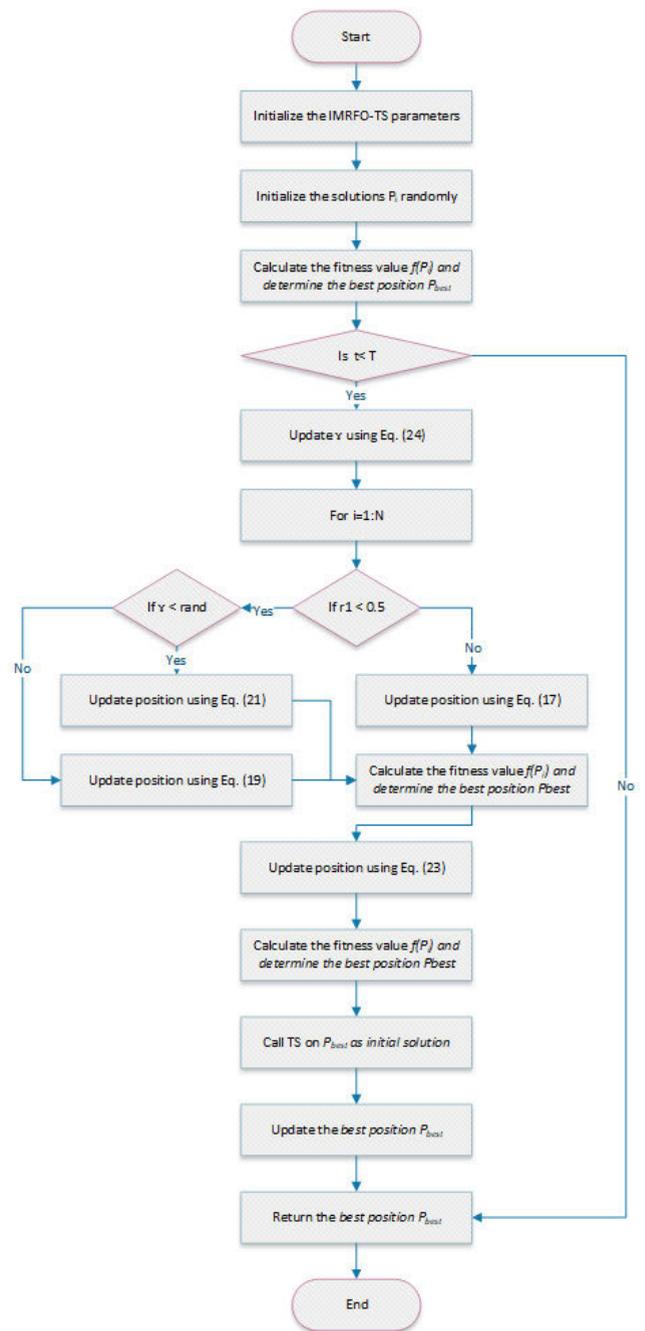


FIGURE 5. Improved manta ray foraging optimization with the Tabu search (IMRFO-TS) flow chart.

of the IMRFO-TS algorithm. MRFO and GWO algorithms present approximately the same best fitness value when U is set at 4. TS, BA, SCA, WOA, RSA algorithms didn't reach their optimal value. Therefore, more UAVs are required for the global solution. FA algorithm, in this scenario, achieves its best fitness value of 0.06 when $U = 10$. However, this value is small compared to the value of 0.28 provided by the IMRFO-TS algorithm. For all algorithms, we can notice that the fitness value is decreasing while the number of UAVs U increases. It can be explained by the fact that the total energy consumed by UAVs is also increasing. Figure 6 displays the

TABLE 4. Algorithms parameters.

Algorithm	Parameter	Value
IMRFO-TS	α	rand
	β	rand
	Number of neighborhood n	50
	Tabu length TL	25
TS	Number of neighborhood n	50
	Tabu length TL	25
BA	Minimum frequency f_{Min}	0
	Maximum frequency f_{Max}	2
	Initial loudness A_0	1
	Initial pulse emission rate r_0	1
	Loudness constant α	0.5
	Emission rate constant γ	0.5
FA	Light absorption coefficient γ	1
	Initial light intensity coefficient I_0	2
	Initial attraction coefficient β_0	2
	Mutation coefficient	0.2
GWO	Mutation coefficient damping ratio	0.98
	Control parameter a_{min}	0
SCA	Control parameter a_{max}	2
	Control parameter r_1	[2,0]
	Control parameter r_2	rand
	Control parameter r_3	rand
WOA	Control parameter r_4	rand
	Control parameter a_{min}	0
MRFO	Control parameter a_{max}	2
	α	rand
RSA	β	rand
	Exploitation adjustment α	0.1
	Exploration adjustment β	0.1

convergence curve given by the algorithms in case of using 4 UAVs for serving 20 users. In this case, we can notice that TS, BA, FA, and WOA algorithms converge quickly in the earlier iterations. This phenomena can be explain by the fact that these algorithms cannot explore new solutions and only consider their sub-optimal solutions as the best. As for SCA and RSA algorithms, we can notice that their convergence behaviour in this case is similar and too slow. MRFO algorithms converges slowly and did not reach its best fitness value yet. The convergence of GWO algorithm is approximately steady during the iterations and its requires more iterations to reach the optimal solution. Our IMRFO-TS algorithm converges to the best fitness cost with an optimal speed. In the earlier iterations, the convergence of our proposed approach is enough quick due its good exploration capabilities improved non linear control strategy. In the last stage of iterations, we can see that the convergence become slow and almost stable. At this stage, the IMRFO-TS searches for better solutions in the area by performing the local search.

To validate the efficiency of the proposed approach, we performed several statistics tests, including the Friedman test as presented in Table 6. From the table, we can see that the proposed IMRFOTS provides the optimal descriptive results, where it gives the highest minimum, maximum, and mean fitness values along the first scenario. Regarding the standards deviation, our method gives good results and closer to the one given by both MRFO and GWO. In this case, we can say that our approach is stable along the test. By analysing the mean rank, we can see that the IMRFOTS gives the best rank compared to the other algorithms. The asymptomatic

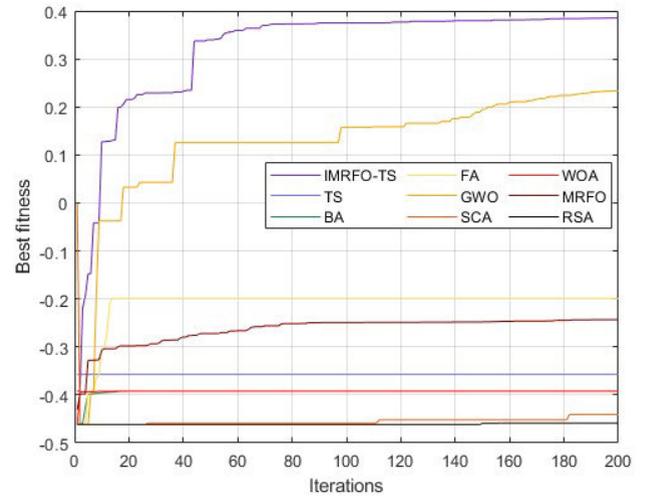


FIGURE 6. Convergence curve in the case of 20 users and 4 UAVs.

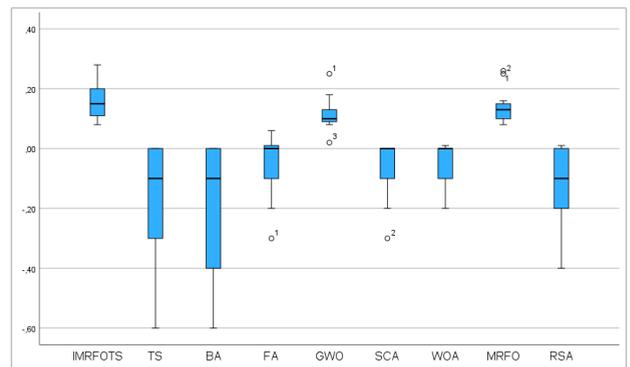


FIGURE 7. The whisker plot of the fitness results obtained in the first case.

signification is less than 0.001 which represents 0.1%. In this case, the signification level indicates that the median of all the algorithms is not equal and null hypothesis is rejected. From these results, we can conclude that our approach clearly outperforms the state-of-the-art algorithms.

Figure 7 illustrates the whisker plots of the fitness provided by the algorithms in the first case of users. From the figure, we can notice that both the median fitness and distribution of the algorithms are not all identical. For the IMRFO-TS algorithm, the statistical data is normally distribution and has no outliers. Both TS and BA algorithms share the median and their data distribution is negative skew. FA, SCA, WOA, and original MRFO data distribution is negative skew and present outliers, except MRFO algorithm. The distribution of both GWO and RSA algorithms is positively skew. In addition, the GWO algorithm has outliers.

The post hoc tests of the fitness for the state-of-the-art algorithms in case of using 20 users, are presented in Table 7. From the table, we can conclude that a statistical significance between IMRFO-TS and TS, BA, FA, SCA, WOA, and RSA algorithms exists based on the significance value and the confidence interval. As for both GWO and MRFO

TABLE 5. Results obtained from different algorithms in the first case.

Results	IMRFO-TS	TS	BA	FA	GWO	SCA	WOA	MRFO	RSA	
<i>Fitness</i>	0.28	-0.64	-0.68	-0.34	0.25	-0.26	-0.26	0.26	-0.42	<i>U</i> = 4
<i>Coverage</i>	95.4	67.9	46.8	69.7	95.2	81.4	83.4	95.4	66.1	
<i>Connectivity</i>	100	81	80	98.5	99.5	87.5	74	89.5	90	
<i>Energy</i>	35.78	59.64	41.55	43.42	36.63	54.6	52.98	38.38	45.42	
<i>Load</i>	0.49	3.44	3.57	2.62	0.56	2.18	2.07	0.43	2.77	
<i>Fitness</i>	0.28	-0.46	-0.68	-0.18	0.18	-0.38	-0.21	0.25	-0.26	<i>U</i> = 5
<i>Coverage</i>	93.6	69.1	55.3	82.6	89.8	65.2	76.7	91.1	82.9	
<i>Connectivity</i>	94	86.8	64.4	95.2	98.8	76	95.2	97.6	80	
<i>Energy</i>	27.49	53.28	40.39	61.37	32.88	39.08	54.21	28.26	59.03	
<i>Load</i>	0.5	2.88	2.65	1.9	0.83	2.53	2.02	0.6	2.07	
<i>Fitness</i>	0.15	-0.3	-0.41	-0.25	0.02	-0.27	-0.19	0.15	-0.4	<i>U</i> = 6
<i>Coverage</i>	87.3	60.8	54	68.7	86.9	55.5	68.5	89.7	34	
<i>Connectivity</i>	100	98	98.67	85	95.67	92	98.67	96	100	
<i>Energy</i>	18.63	38.85	33.07	53.3	29.78	33.1	42.75	21.09	17.72	
<i>Load</i>	1.12	2.41	2.85	1.99	1.44	2.24	2	0.98	2.75	
<i>Fitness</i>	0.23	-0.34	-0.68	-0.1	0.09	-0.14	-0.16	0.13	-0.34	<i>U</i> = 7
<i>Coverage</i>	91.7	49.1	55.5	68.5	83	80.5	60.2	83.8	45.8	
<i>Connectivity</i>	99.71	90	100	98.86	100	96.57	99.14	99.71	94.57	
<i>Energy</i>	22.75	41.77	33.21	47.51	23	54.02	34.96	21.80	35.09	
<i>Load</i>	0.76	2.33	2.24	1.6	1.22	1.79	1.87	1.11	2.41	
<i>Fitness</i>	0.18	-0.3	-0.27	-0.09	0.08	-0.14	-0.13	0.16	-0.23	<i>U</i> = 8
<i>Coverage</i>	91.3	52.7	64	68.7	84.6	63.9	67.5	91.2	67	
<i>Connectivity</i>	100	85.25	99.75	100	99.75	96.25	99.5	100	73.75	
<i>Energy</i>	23.61	41.05	41.86	50.55	24.48	33.69	49.39	23.87	30.09	
<i>Load</i>	0.95	2.16	2.31	1.54	1.29	1.82	1.7	1.03	2.03	
<i>Fitness</i>	0.12	-0.21	-0.16	-0.05	0.08	-0.12	-0.12	0.15	-0.17	<i>U</i> = 9
<i>Coverage</i>	88.3	79.3	89.5	71.9	85.9	64	55.8	92.9	43.8	
<i>Connectivity</i>	100	97.56	100	100	99.78	96.89	99.78	100	99.56	
<i>Energy</i>	20.75	51.32	51.04	56.58	24.04	35.68	28.80	24.39	30.96	
<i>Load</i>	1.2	2.1	2.06	1.34	1.31	1.73	1.75	1.07	1.82	
<i>Fitness</i>	0.08	-0.13	-0.11	0.06	0.13	-0.08	-0.02	0.08	-0.11	<i>U</i> = 10
<i>Coverage</i>	79.8	74.8	81.7	85.1	86.1	63.4	73.2	74.5	54.10	
<i>Connectivity</i>	100	100	100	100	100	97.8	100	100	100	
<i>Energy</i>	14.39	40.78	40.57	50.22	23.12	29.03	37.89	12.94	27.35	
<i>Load</i>	1.34	1.86	1.11	1.12	1.87	1.66	1.42	1.31	1.69	
<i>Fitness</i>	0.11	-0.11	-0.14	0.05	0.12	-0.07	-0.06	0.13	-0.14	<i>U</i> = 11
<i>Coverage</i>	79.8	66.5	69.3	76.1	79.4	60.7	49.7	74.8	46.6	
<i>Connectivity</i>	100	92.18	100	100	100	98.18	100	100	100	
<i>Energy</i>	17.34	42.74	48.81	53.23	21.18	29.92	30.14	14.41	35.25	
<i>Load</i>	1.15	1.6	1.78	1.04	1.1	1.59	1.43	1.09	1.7	
<i>Fitness</i>	0.1	-0.08	-0.11	-0.04	0.11	-0.07	-0.03	0.16	-0.13	<i>U</i> = 12
<i>Coverage</i>	65.4	46.9	70.3	64.2	73	44.1	67.4	81.5	14.5	
<i>Connectivity</i>	100	99.67	100	100	100	99.67	98.83	100	100	
<i>Energy</i>	14.28	37.88	50.12	30.3	21.42	21.91	43.39	23.07	5.48	
<i>Load</i>	1.13	1.42	1.65	1.49	1.05	1.49	1.35	0.95	1.61	
<i>Fitness</i>	0.15	-0.06	-0.08	-0.01	0.12	-0.05	0.01	0.09	-0.06	<i>U</i> = 13
<i>Coverage</i>	79.2	38.4	32.7	71.4	74.8	38.5	63.7	69.5	53.1	
<i>Connectivity</i>	100	99.69	100	99.54	99.85	99.07	99.85	100	99.54	
<i>Energy</i>	25.94	29.61	34.91	43.18	23.1	17.98	34.4	22.09	27.99	
<i>Load</i>	0.94	1.32	1.32	1.3	1.03	1.41	1.22	1.1	1.48	
<i>Fitness</i>	0.11	-0.08	-0.09	0.03	0.09	-0.04	0	0.13	-0.03	<i>U</i> = 14
<i>Coverage</i>	63.9	66.5	35.4	72	61.2	29.2	48.8	76.4	52	
<i>Connectivity</i>	100	99.43	93.14	99.57	93.14	99.43	98.86	100	100	
<i>Energy</i>	13.83	40.77	35.77	53.4	20.61	10.30	27.55	21.34	25.11	
<i>Load</i>	1.09	1.17	1.3	1.06	1.05	1.34	1.22	1.03	1.38	
<i>Fitness</i>	0.17	-0.07	-0.08	0.01	0.13	0	0.01	0.12	0.01	<i>U</i> = 15
<i>Coverage</i>	89	62.8	51.1	62.5	73.9	41.3	65.7	72.3	53.9	
<i>Connectivity</i>	100	99.47	99.87	100	99.87	99.2	99.87	100	100	
<i>Energy</i>	33.83	44.19	42.25	48.93	20.31	16	44.91	20.19	30.17	
<i>Load</i>	0.87	1.47	1.41	1.1	1.02	1.24	1.15	1.05	1.21	
<i>Fitness</i>	0.2	-0.03	-0.03	0.03	0.17	0.01	0.04	0.12	0.01	<i>U</i> = 16
<i>Coverage</i>	89.2	72.1	75	55.3	83.4	43.4	63.7	61.3	78.6	
<i>Connectivity</i>	100	99.62	100	100	100	99.87	99.35	99.87	100	
<i>Energy</i>	20.73	40.18	37.43	36.46	20.21	19.64	29.9	14.74	38.03	
<i>Load</i>	0.9	1.44	1.5	1.1	0.96	1.19	1.15	1	1.35	

algorithms, no statistical significance exists since their confidence interval contains zero and the significance value exceed the 0.5. Therefore, the null hypothesis is accepted for both GWO and MRFO. Remaining the rest algorithms, it is rejected.

Figure 10 illustrates the mean coverage rate of the algorithms using various number of UAVs. In most cases, the

IMRFO-TS algorithm provides the best coverage quality. The best coverage quality given by IMRFO-TS is 95.4% achieved when $U = 4$. In this case, both GWO and MRFO provided their best rates which are 95.4% and 95.2%, respectively. TS, SCA, WOA, and RSA algorithms provide a poor coverage quality less than 85% which makes them inappropriate for this scenario. BA and FA achieved 89.5% and 85.1% of users

TABLE 6. The Friedman tests of the fitness cost in the first case.

	Results	IMRFO-TS	TS	BA	FA	GWO	SCA	WOA	MRFO	RSA
Descriptive statistics	N	13	13	13	13	13	13	13	13	13
	Minimum	0.08	-0.6	-0.6	-0.3	0.2	-0.3	-0.2	0.08	-0.4
	Maximum	0.28	0	0	0.06	0.25	0	0.01	0.26	0.01
	Mean	0.17	-0.18	-0.21	-0.04	0.11	-0.08	-0.06	0.14	-0.14
	STD	0.07	0.19	0.25	0.11	0.05	0.10	0.08	0.06	0.15
Rank	Mean rank	8.46	2.31	2.31	4.96	7.5	3.85	4.54	8.04	3.04
Statistic tests	N	13								
	Chi-square	91.27								
	Df	8								
	Asymp. sig	< 0.001								

TABLE 7. The post hoc test of the mean fitness of each algorithm in the fifth case.

		Multiple Comparison				
		Dependent variable: Fitness				
Algorithm (I)	Algorithm (J)	Mean difference (I-J)	SE	Sig	95 % confidence interval	
					Lower bound	Upper bound
IMRFO-TS	TS	0.38	0.5	< 0.001	0.28	0.49
	BA	0.44	0.53	< 0.001	0.33	0.54
	FA	0.23	0.53	< 0.001	0.13	0.34
	GWO	0.5	0.53	0.4	-0.06	0.15
	SCA	0.29	0.53	< 0.001	0.18	0.4
	WOA	0.25	0.53	< 0.001	0.15	0.36
	MRFO	0.02	0.53	0.74	-0.09	0.12
	RSA	0.34	0.53	< 0.001	0.23	0.45

coverage, respectively in cases $U = 9$ and $U = 10$, respectively. By varying the number of UAVs, it can be noticed that the coverage quality is decreasing due to the coverage radius minimization related to UAVs' heights.

Figure 11 depicts the mean connectivity rate obtained by the algorithms under different number of UAVs. It can be seen that the proposed IMRFO-TS algorithm improves significantly the connectivity rate compared to the other algorithms. Starting from $U = 8$, the proposed approach achieves full connectivity between UAVs. When $U < 8$, the connectivity rate provided by the IMRFO-TS is considered high since it is varied between 94% and 100%. After IMRFO-TS, MRFO and GWO algorithms provide good connectivity quality more than 89%. Both of them achieve the full connectivity 5 and 7 times, respectively. Both TS and WOA algorithms offer more than 85% of UAV connectivity starting from $U = 5$. BA, SCA, and RSA are considered optimal in terms of UAV connectivity when $U \geq 6$. The connectivity rate provided by FA is good and more than 85% regardless the number of UAVs.

Figure 8 illustrates the mean energy consumed by different UAVs in case of 20 users. We can see that the power source of UAVs is decreasing when the number of UAVs is rising up. It can be justified by the fact that algorithms tend to minimize the cost of using UAVs and improve their efficiency by reducing energy cost. The total energy consumed by UAVs is around 20% and continuously decreasing until it reached its minimum value of 13.83% when $U = 14$. The total energy consumed in case of using IMRFO-TS algorithm solutions is reasonable and acceptable since the algorithm is improving the coverage rate. MRFO and GWO presented close results

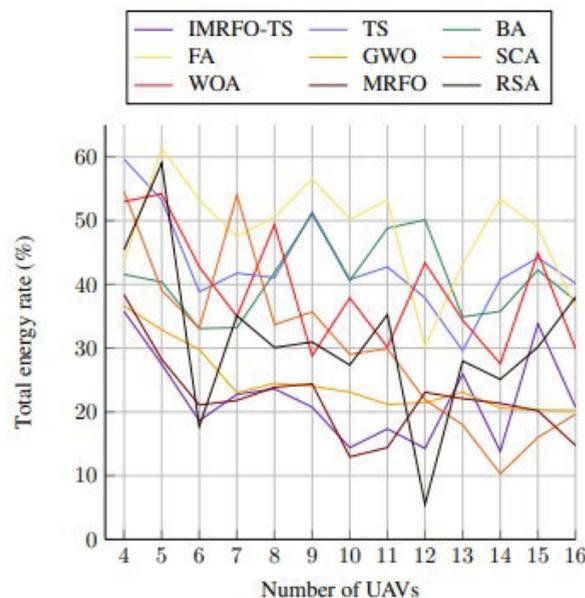


FIGURE 8. The average total energy consumption in first case.

and their total energy values are good. IMRFO-TS, MRFO, and GWO make a good balance between the energy constraint and the coverage objective. The solutions found by SCA algorithm focused on the energy aspect with less consideration of the coverage quality. The energy values provided by RSA are little high and can be acceptable for $U \geq 9$. TS, BA, FA, and WOA showed poor performance in all cases. Their total energy consumption is considered high and not efficient which makes them inappropriate for this case of users.

TABLE 8. The Friedman tests of the fitness cost in the second case.

	Results	IMRFO-TS	TS	BA	FA	GWO	SCA	WOA	MRFO	RSA	
Descriptive statistics	N	13	13	13	13	13	13	13	13	13	
	Minimum	-0.28	-1.88	-1.88	-1.59	-0.5	-1.42	-1.22	-0.42	-1.99	
	Maximum	0.23	-0.63	-0.68	-0.36	0.2	-0.49	-0.07	0.2	-0.56	
	Mean	-0.13	-1.08	-1.11	-0.79	-0.23	-0.88	-0.74	-0.25	-1.07	
	STD	0.15	0.43	0.38	0.35	0.23	0.27	0.31	0.18	0.45	
Rank	Mean rank	8.69	2.27	1.38	5.62	7.54	4.31	4.92	7.69	2.58	
Statistic tests	N						13				
	Chi-square						94.86				
	Df						8				
	Asymp. sig						< 0.001				

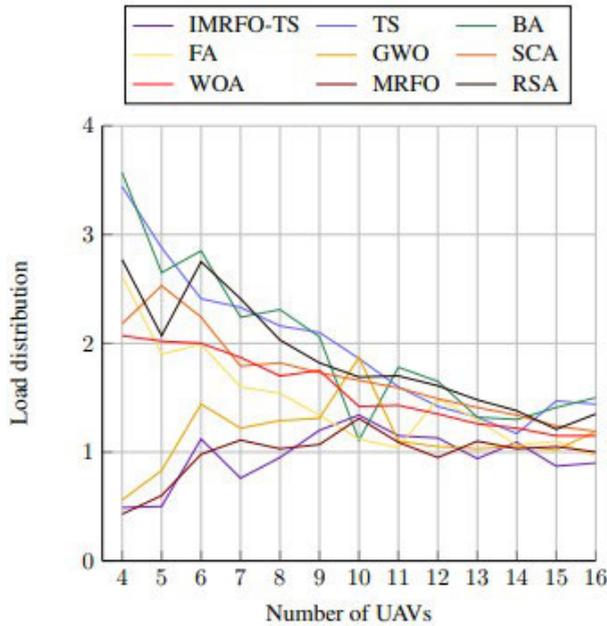


FIGURE 9. The average load distribution in the first case.

Figure 9 presents the mean load distribution. It can be seen that all algorithms provided good results regarding this metric. In most cases, the mean load distribution is around 1 which is optimal since it represents 1.67% of total users. IMRFO-TS and MRFO in this scenario provided closer results. But the proposed IMRFO-TS outperforms MRFO algorithm by one case difference. Comparing the algorithms, BA provided the highest load value. RSA and TS are ranked after BA, respectively. Fa results are less than BA, RSA, and TS results, but more than SCA and WOA results, respectively. Finally, both IMRFO-TS and MRFO algorithms are the most suitable methods for preserving UAVs resources according to the load distribution results.

B. CASE OF 60 USERS

Table 15 details the results obtained of 60 users under various number of UAVs. It can be noticed that the proposed IMRFO-TS outperforms the compared algorithm by giving the highest fitness value in most cases. The IMRFO-TS algorithm achieves its maximum value of 0.23 when the

number of UAVs is set to 4. MRFO and GWO algorithms provide an equal maximum fitness value of 0.2 using 5 and 6 UAVs respectively. The fitness value of TS, BA, SCA, WOA, and RSA algorithms is continuously increasing by rising up the number of UAVs. Consequently, these algorithms didn't reach the optimal value yet and need more UAVs. Figure 12 shows the convergence curve of the algorithms in the case of 4 UAVs are used. As expected, The convergence of TS, BA, and FA algorithms is efficient in terms of speed. However, their convergence can not be considered optimal regarding the rate since their fitness results are not optimal. The convergence of RSA, SCA, and MRFO algorithms is slow in this case due to their weakness in exploration. As for WOA algorithm, we can see that its convergence is not steady in this case and its behaviour changes among the iterations, which demonstrates its inability to achieve the balance between the exploration/exploitation search. Regarding the GWO algorithm, we can see that it is still performing the exploration search and additional iterations are required in this case. From the figure, we can notice that the best fitness value is offered by our IMRFO-TS approach, which demonstrates its efficiency in terms of convergence rate. As for the convergence speed, we can consider it optimal due to the fact that the IMRFO-TS algorithm is enough fast to identify the area where the optimal solution is located. Moreover, our approach shows slow convergence to search for better solutions in the area.

Table 8 describes the statistical test results of the algorithms in the second case. It can be seen that, the IMRFO-TS algorithm gives the highest results regarding the minimum, maximum, and means fitness. Therefore, our approach is the best in terms of finding the optimized UAV positions. Moreover, the IMRFO-TS is the most stable in this case, which is proved by the standard deviation results. In this scenario, the best mean rank is given by our approach, which demonstrates its efficiency. As for the significance level, it less than 0.1% in this case. Therefore, the null hypothesis is rejected and the difference between the algorithms in terms of performance exists.

Based on the data presented in Table 8 we created a whisker plot of the fitness results for the algorithms as presented in Figure 13. In this case the median fitness given by the algorithms is not identical at all. As for the data distribution, IMRFO-TS, GWO, and RSA algorithms have positive skew distribution. The other algorithms present negative skew

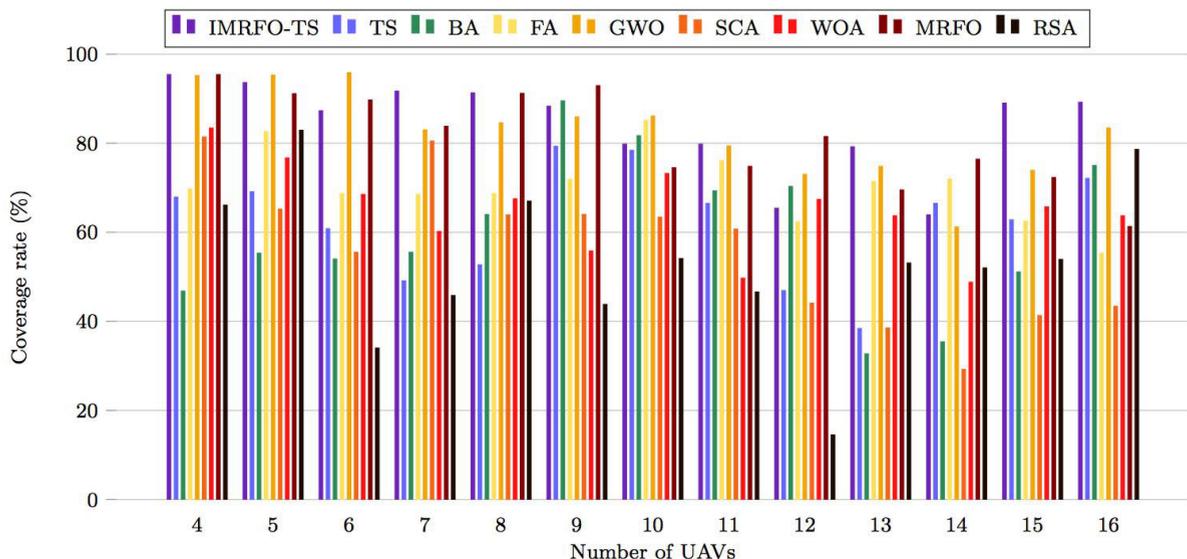


FIGURE 10. The average coverage rate in the first case.

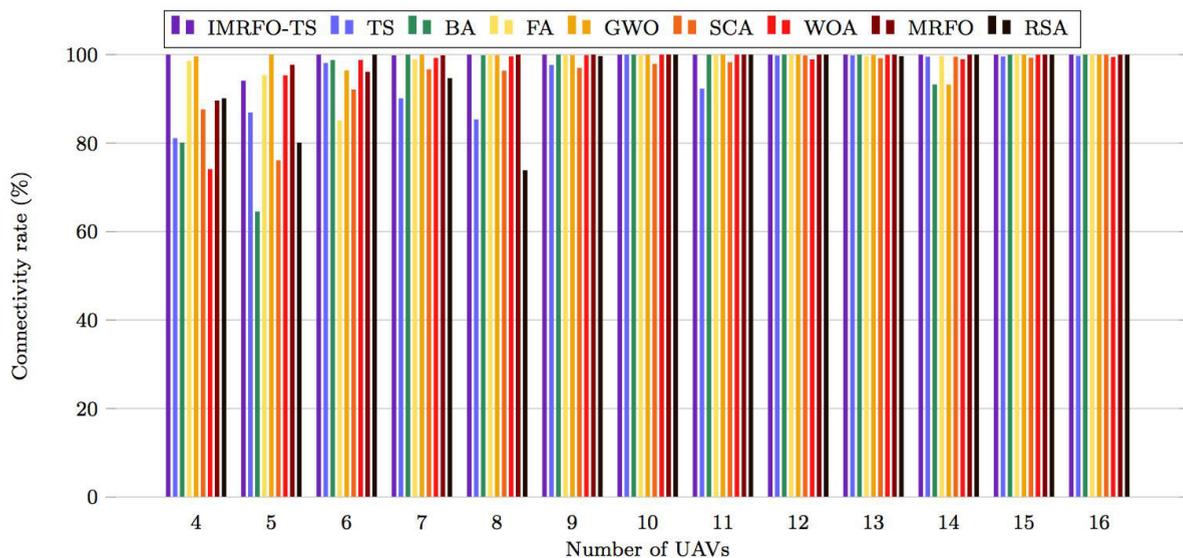


FIGURE 11. The average connectivity rate in the first case.

distribution. The IMRFO-TS, GWO, WOA, and MRFO algorithms present outliers in this scenario. Table 9 summarized the post hoc of results of the algorithms in case of using 60 users in the area. The table shows that a statistical significance with algorithms exist. The significance value for TS, BA, FA, SCA, WOA, and RSA algorithms represent 0.1%, which is less than 5%. In addition, the confidence interval is strictly positive. Therefore, the statistical significance exist and the null hypothesis is rejected. The significance value for both GWO and MRFO algorithms exceed the 0.5. In this case, the statistical significance does not exist and the null hypothesis is accepted.

Regarding the coverage metric, IMRFO-TS gives the best coverage quality when $U = 4$ as shown in Figure 16. IMRFO-TS is close to cover all users with a coverage rate of 98.8%. After IMRFO-TS, the MRFO algorithms provides a good coverage rate of 98.5% for $U = 8$. GWO and WOA provide their best coverage rate of 98.6% and 93.8%, respectively for $U = 4$. The coverage quality provided by TS, BA, FA, SCA, and RSA algorithms is poor and less than 85%. By varying the number of UAVs, the coverage quality is decreased due to the impact of energy resources. IMRFO-TS ensures a coverage quality more than 85% when the number of UAVs is less than 9. GWO and MRFO can provide that

TABLE 9. The post hoc test of the mean fitness of each algorithm in the second case.

Multiple Comparison						
Dependent variable: Fitness						
Algorithm (I)	Algorithm (J)	Mean difference (I-J)	SE	Sig	95 % confidence interval	
					Lower bound	Upper bound
IMRFO-TS	TS	0.95	0.13	< 0.001	0.7	0.86
	BA	0.98	0.13	< 0.001	0.73	1.23
	FA	0.66	0.13	< 0.001	0.41	0.91
	GWO	0.09	0.13	0.45	-1.55	0.34
	SCA	0.75	0.13	< 0.001	0.5	1
	WOA	0.61	0.13	< 0.001	0.36	0.86
	MRFO	0.12	0.13	0.36	-0.13	0.37
	RSA	0.94	0.13	< 0.001	0.69	1.19

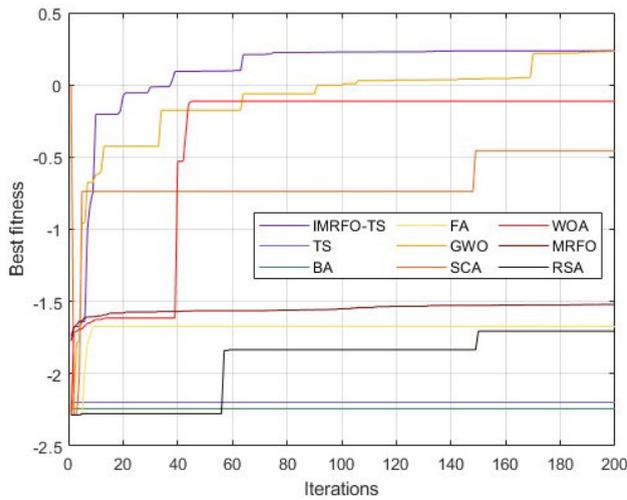


FIGURE 12. Convergence curve in the case of 60 users and 4 UAVs.

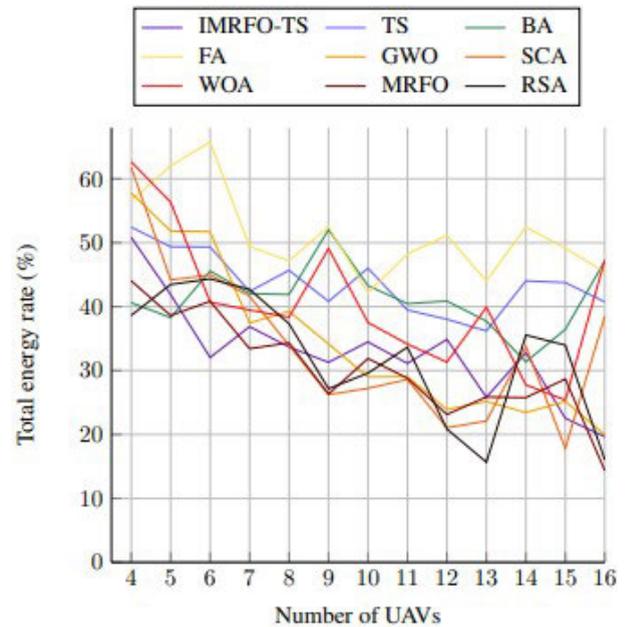


FIGURE 14. The average total energy consumption in the second case.

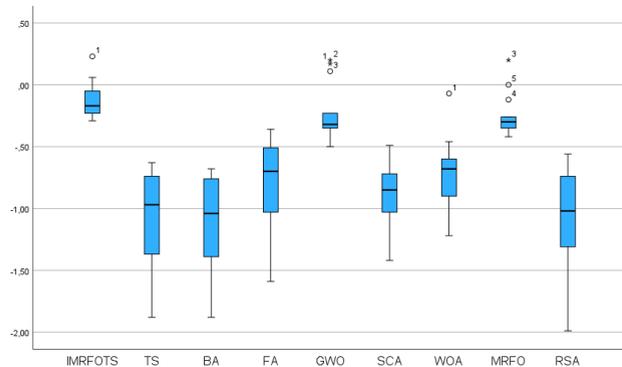


FIGURE 13. The whisker plot of the fitness results obtained in the second case.

quality until $U = 8$. WOA gives only a good quality when $U = 4$. The quality of the other algorithms is not acceptable and continues to decrease.

Figure 17 illustrates the mean connectivity rate. It can be noticed that the connectivity quality increases as the number of UAVs increases. In fact, adding more UAVs, gives rise to the size of the network and produces more mesh links between them. For the IMRFO-TS algorithm, the use of 4 and

5 UAVs is not the optimal one. Starting from $U = 6$, the connectivity quality given by IMRFO-TS is increased from 97.43% to 100% where it achieves the full connectivity 4 times when $U = \{9, 20, 14, 16\}$. After BA algorithm in case of $U = \{5, 7, 9, 11, 13\}$. GWO and FA algorithms provide a good connectivity rate greater than 85%. WOA and MRFO algorithm improve the connectivity quality when the number of drones is greater than 6. Tabu search starts to give good connectivity results starting from $U = 7$. SCA and RSA reach more than 85% of connectivity when $U > 8$. The connectivity quality provided by BA is not stable when $U < 9$.

Figure 14 illustrates the mean energy provided by the algorithms. We can see that the energy is decreasing when the number of UAVs is increasing. Algorithms provide low heights UAVs positions that minimize the energy. The IMRFO-TS archives the best energy value of 19.65% when $U = 16$ after MRFO and RSA with the value of 14.35% and 16.04% respectively. Except the case of $U = \{4, 5\}$, IMRFO-TS provides mean energy around 30%. In the scenario,

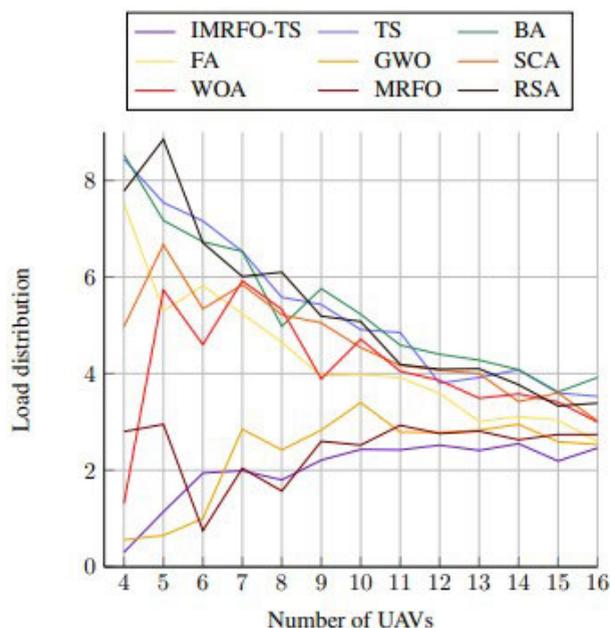


FIGURE 15. The average load distribution in the second case.

solutions given by both SCA, and RSA tend to optimize the energy metric more than the coverage. starting from $U = 9$, we can see that both of them give a good energy percentage. But the coverage quality is less than 50% in most cases. Consequently, SCA and RSA are not able to balance between different objectives. IMRFO-TS provides the best balance between coverage and energy. It works on minimizing the energy while keeping a coverage quality over 70% in most cases. MRFO and GWO algorithms also give good balance after the IMRFO-TS algorithm. The best energy results given by WOA algorithm are achieved when $U = \{14, 15\}$, but without a good balance. The results given by TS, BA, and FA algorithms are not considered optimal. The mean energy rate is high compared to the others.

Figure 15 represents the mean load distribution. We can notice that the IMRFO-TS algorithm provides the smallest load distribution value in most cases. Therefore, our proposed algorithm splits approximately equal number of users between UAVs which preserves better UAVs resources. After IMRFO-TS algorithm, MRFO and GWO algorithms provide good results regarding the load. The load results given by TS, BA, FA, SCA, WOA, and RSA algorithms are high but acceptable since it is less than 8% of total users in most cases.

C. CASE OF 100 USERS

Table 16 summarized the obtained results in case of 100 users using different number of UAVs. According to the table, IMRFO-TS algorithm provided the best fitness value in most cases as expected. The proposed method achieved the optimal fitness of 0.19 which is the greatest value in case of $U = 4$. In the same case, both GWO and MRFO algorithms provided their best fitness value of -0.08 and -0.18, respectively. The other algorithms did not converge to their optimal value

and their mean fitness is improving by adding more UAVs. IMRFO-TS, GWO, and MRFO algorithms need few number of UAVs to converge unlike TS, BA, FA, SCA, WOA, and RSA algorithms. Consequently, IMRFO-TS, GWO, and MRFO are the most optimal regarding the deployment cost of UAVs. Figure 18 illustrates the best fitness over iterations given by the algorithms in the case of 4 UAVs. As Shown in the figure, the convergence of TS, BA, and SCA algorithms is stable over the iterations due to their lack in finding new solutions. RSA algorithms convergence slowly and need more iterations. FA and WOA converge in this case with the same manner, where both of them reach their best in the earlier iterations. As for GWO and MRFO algorithms, we can see that their convergence is little stable by the end of iterations, which demonstrates their limitation in exploitation. In this case, the proposed IMRFO-TS approach shows the best performance in terms of both speed and rate due to non linear control and local search mechanisms.

Table 10 details the statistical results given by the algorithms in the third case. We can notice from the table that the best descriptive statistic results is given by the IMRFO-TS algorithm. Our approach gives the highest minimum, maximum, and mean values. As for the standard deviation, the original MRFO algorithm gives the smallest value. However, the STD given by our approach is close to the one given by MRFO. Therefore, the stability our approach is optimal and proved the robustness of our approach in avoiding trapping into local optima. Regarding the mean rank, our approach is classified the first and the rank gap with other algorithms is clearly noticed. By analysing the asymptomatic significance, we can see that it is small and $< 0.1\%$. Therefore, the null hypothesis is rejected also in this case.

Figure 19 displays the whisker plot of the fitness results in case of using 100 users. From the figure, the median difference of the algorithms is clearly noticed in this case. In addition, The data distribution is not the same for all the algorithms. The IMRFO-TS, BA, FA, SCA, MRFO, and RSA algorithms show a negative skewed distribution. The distribution of the other algorithms is positively skewed. Moreover, the FA, GWO, and WOA algorithms present outliers.

Table 11 details the post hoc results using the least significant difference (LSD) test for the algorithms. The table reveals that there is a significance difference between the IMRFO-TS algorithm and the rest. By analysing the significant value, we can see that it is less than 0.5% for all algorithms, except both GWO and MRFO algorithms. In this case, we can say that, only GWO and MRFO algorithms are not significant. In addition, the confidence interval for the mean difference between IMRFO-TS and both GWO and MRFO algorithms contains zero. Thus, the null hypothesis that there is not statistical significance between them is rejected. As for the rest, the significant value is less than 5% and the confidence interval is positive. Therefore, the statistical significance exists.

Figure 24 illustrated the mean coverage rate in a scenario where 100 users are deployed. It can be noticed that the

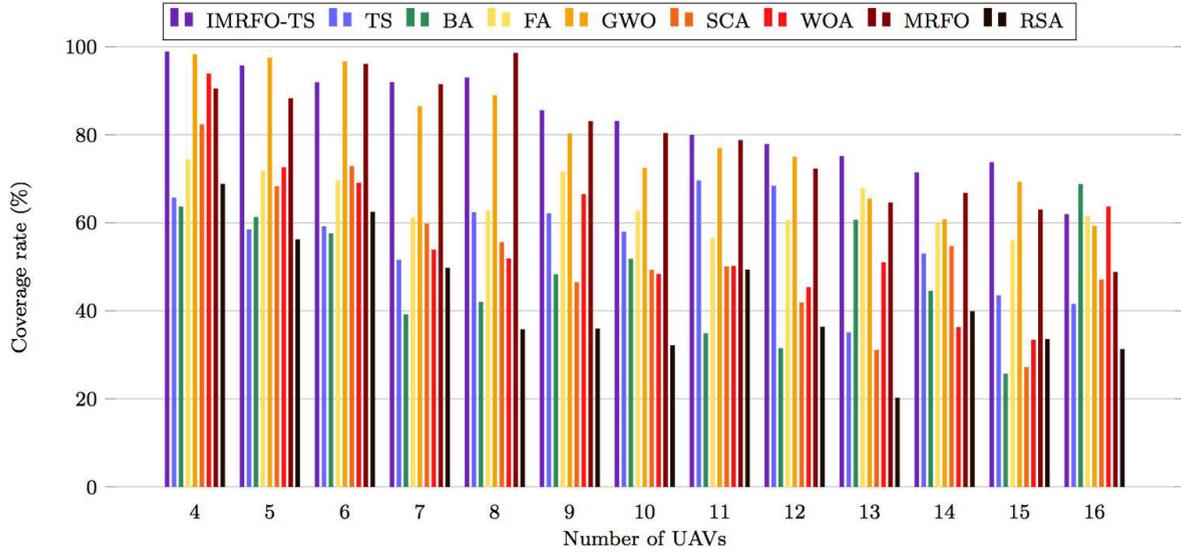


FIGURE 16. The average coverage rate in the second case.

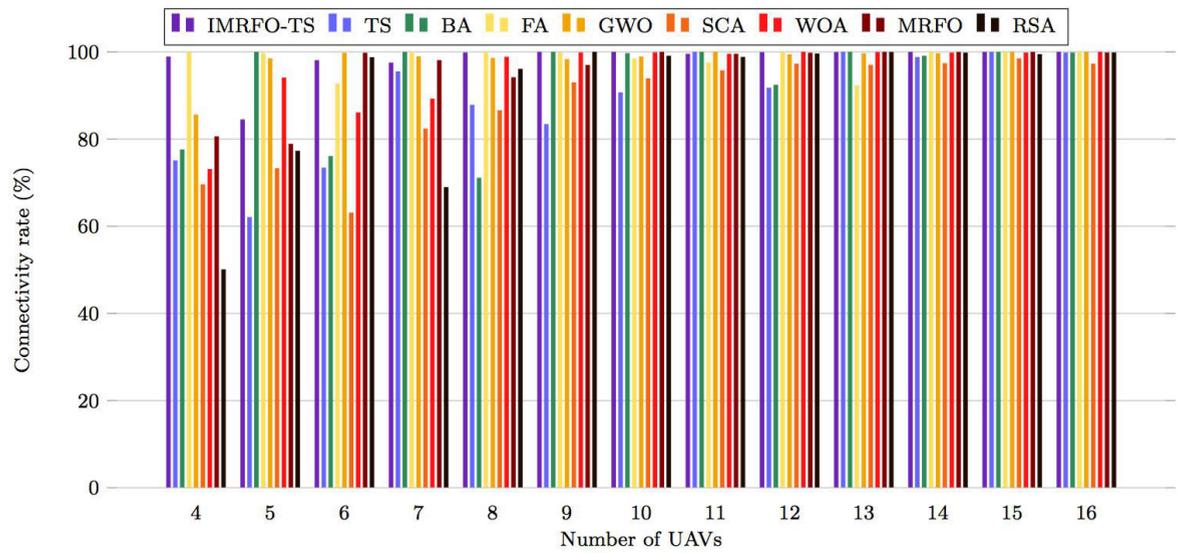


FIGURE 17. The average connectivity rate in the second case.

TABLE 10. The Friedman tests of the fitness cost in the second case.

Results		IMRFO-TS	TS	BA	FA	GWO	SCA	WOA	MRFO	RSA	
Descriptive statistics	N	13	13	13	13	13	13	13	13	13	
	Minimum	-0.74	-3.05	-3.6	-2.85	-0.84	-2.75	-2.74	-0.81	-3	
	Maximum	0.19	-1.22	-1.14	-0.9	-0.07	-1.09	-0.91	-0.18	-1.16	
	Mean	-0.4	-1.98	-2.04	-1.5	-0.61	-1.73	-1.49	-0.49	-1.89	
	STD	0.29	0.64	0.83	0.54	0.25	0.53	0.5	0.2	0.63	
Rank	Mean rank	8.85	1.81	1.96	5.19	7.23	3.73	5.42	7.92	2.88	
Statistic tests	N						13				
	Chi-square						93.97				
	Df						8				
	Asymp. sig						< 0.001				

TABLE 11. The post hoc test of the mean fitness of each algorithm in the third case.

Multiple Comparison						
Dependent variable: Fitness						
Algorithm (I)	Algorithm (J)	Mean difference (I-J)	SE	Sig	95 % confidence interval	
					Lower bound	Upper bound
IMRFO-TS	TS	1.58	0.21	< 0.001	1.17	1.99
	BA	1.64	0.21	< 0.001	1.23	2.05
	FA	1.1	0.21	< 0.001	0.69	1.51
	GWO	0.21	0.21	0.31	-1.2	0.62
	SCA	1.33	0.21	< 0.001	0.92	1.74
	WOA	1.1	0.21	< 0.001	0.68	1.51
	MRFO	0.1	0.21	0.64	-0.31	0.51
	RSA	1.5	0.21	< 0.001	1.08	1.9

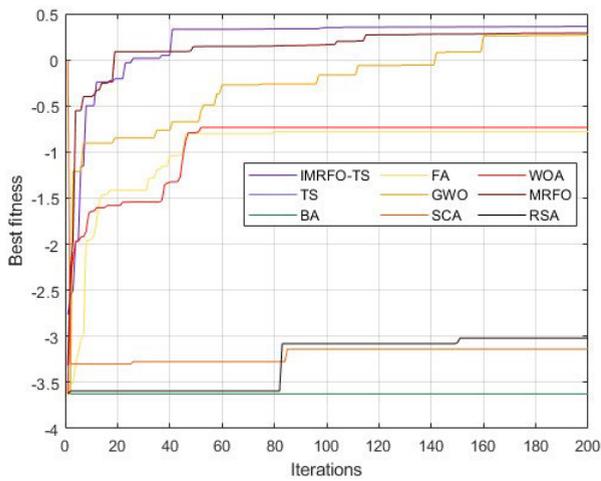


FIGURE 18. Convergence curve in the case of 100 users and 4 UAVs.

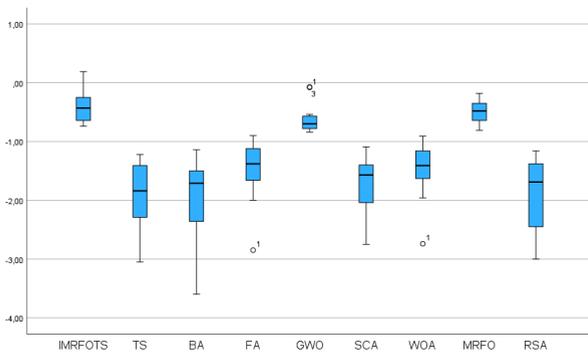


FIGURE 19. The whisker plot of the fitness results obtained in the third case.

proposed IMRFO-TS provides the best coverage quality in most cases compared to the other algorithms. When $U < 10$, IMRFO-TS offers more than 88% of users coverage. Moreover, the coverage rate increases until 99.1% for $U = 4$. In other cases, the coverage rate given is acceptable and around 70%. Similar to IMRFO-TS algorithm, MRFO covers more than 88% of users when $U < 10$. However, MRFO algorithm in this case of users outperforms GWO algorithm by giving better coverage quality. The coverage rate provided by GWO is more than 85% for $U < 9$. A part from

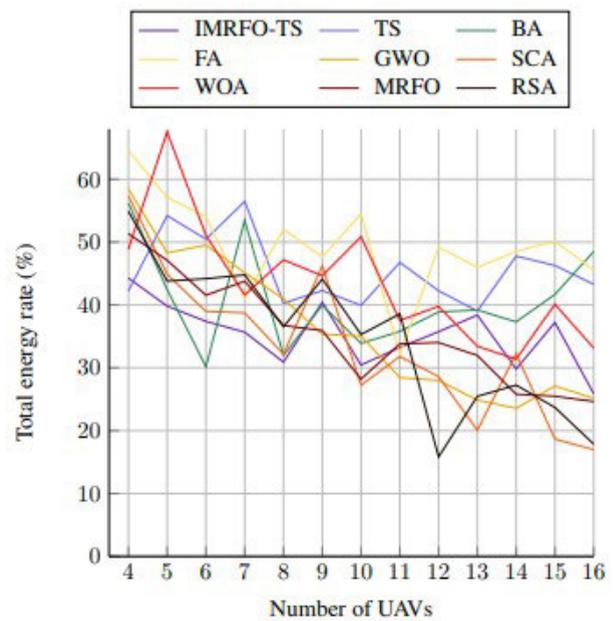


FIGURE 20. The average total energy consumption in the third case.

this number of UAVs, MRFO, and GWO algorithms have coverage rates of around 70% and 60%, respectively. The coverage results given by TS, BA, FA, SCA, WOA, and RSA algorithms are poor and around 40%.

Figure 25 depicts the mean UAVs connectivity in case of 100 users. It can be seen that IMRFO-TS algorithm provides the best connectivity rate in most cases. In case of $U \leq 6$, IMRFO-TS method achieves more than 98% of UAVs connectivity and reaches the full connectivity for $U = \{14, 15\}$. The other algorithms also presented good connectivity results and greater than 85% in most cases. However, the IMRFO-TS algorithm outperforms the other meta-heuristics.

Regarding the energy metric, Figure 20 presents the mean total energy consumed by different number of UAVs serving 100 users. We can notice that the total energy consumed when IMRFO-TS is used is acceptable and around 35%. IMRFO-TS gives its best energy value of 25.86% for $U = 16$. From $U = 12$, SCA and RSA algorithms improve the energy consumption over other metrics where their both give their best of 16.98% and 15.83%, respectively. MRFO presents similar results to IMRFO-TS algorithm. However, GWO results are

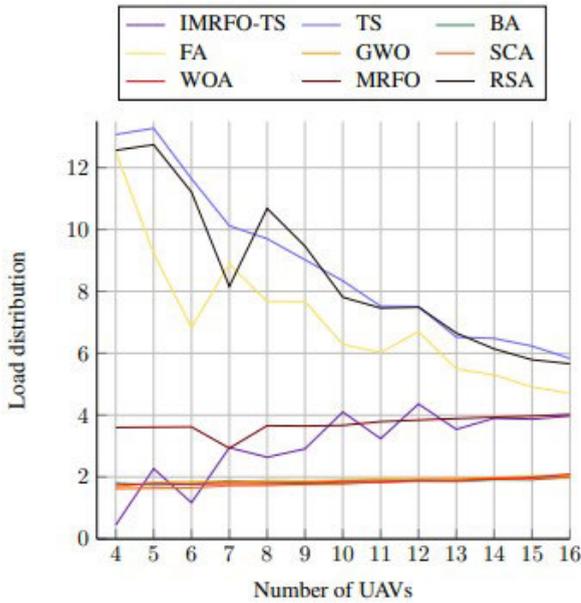


FIGURE 21. The average load distribution in the third case.

better and around 30% of total energy used. The results given by TS, BA, and WOA are approximately similar and around 40% which is high and inefficient. Using FA algorithm, 50% of energy is used and unacceptable practically.

Figure 21 illustrates the average load distribution given by different UAVs for 100 users. It can be seen that the most optimal load is provided by the proposed approach for $U = 4$. For $U \neq 4$, the load increases from 1.17 to 4.36 which represent a gap of 4.5%. This gap is considered optimal and acceptable in view of the coverage quality offered. In this scenario, GWO provides good load results and MRFO offers close results to the one given by IMRFO-TS. The obtained load values by BA, SCA, and WOA algorithms are small and good. But, we cannot consider them optimal because using these algorithms most of users and not covered which explain the small load values. As for TS, FA, and RSA algorithms the gap of user distribution between UAVs is remarkable. But, it can be acceptable since its represents less than 10% in most cases.

D. CASE OF 140 USERS

Table 14 shows the obtained results by the meta-heuristics in case of 140 users. As it can be noticed, the proposed IMRFO-TS outperforms state-of-the-art meta-heuristics in most cases. IMRFO-TS gives the most fitness value of 0.3 for $U = 7$. In this case, GWO and MRFO algorithms provide the best fitness value of 0.17 and 0.16, respectively. As expected, the other algorithms did not converge yet to their optimal values and need more UAVs. Figure 26 illustrates the convergence curve of the algorithms in case of using 4 UAVs. From the figure, we can notice that TS, BA, and FA algorithms converge to their best fitness in the earlier iterations. In this case, these algorithms trapped into local optima and not able to explore new solutions. As for SCA and RSA algorithms, we can see that their convergence is slow. both of them

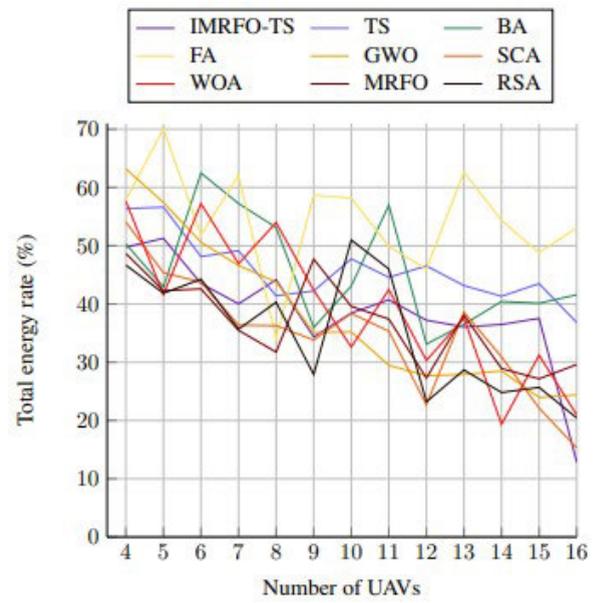


FIGURE 22. The average total energy consumption in the fourth case.

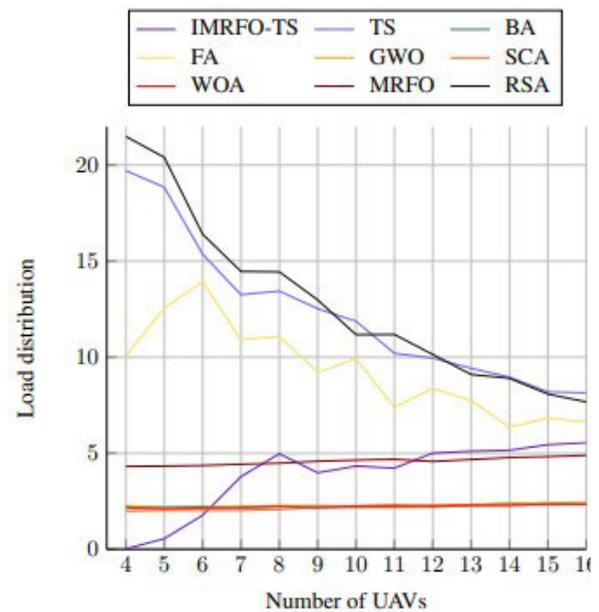


FIGURE 23. The average load distribution in the fourth case.

reach their best solution in the last iterations. By analysing the convergence of GWO algorithm, we can see that it is relatively slow and it requires more iterations in this case. As for WOA algorithm, it converges to a good fitness value with efficient speed. The convergence behaviour of original MRFO algorithm is similar to WOA algorithm. However, in terms of efficiency, WOA performs better than MRFO algorithm. In this case, the convergence speed of our proposed approach is sufficiently fast due to the good balance between the exploration and exploitation phases. Moreover, in terms of efficiency, our approach outperforms state-of-the-art meta-heuristics.

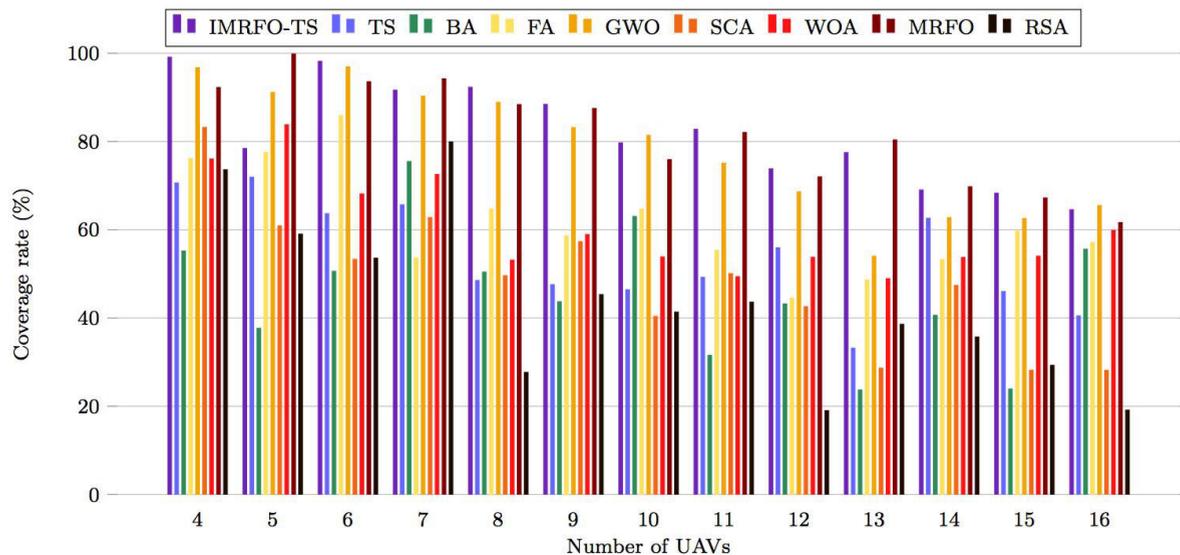


FIGURE 24. The average coverage rate in the third case.

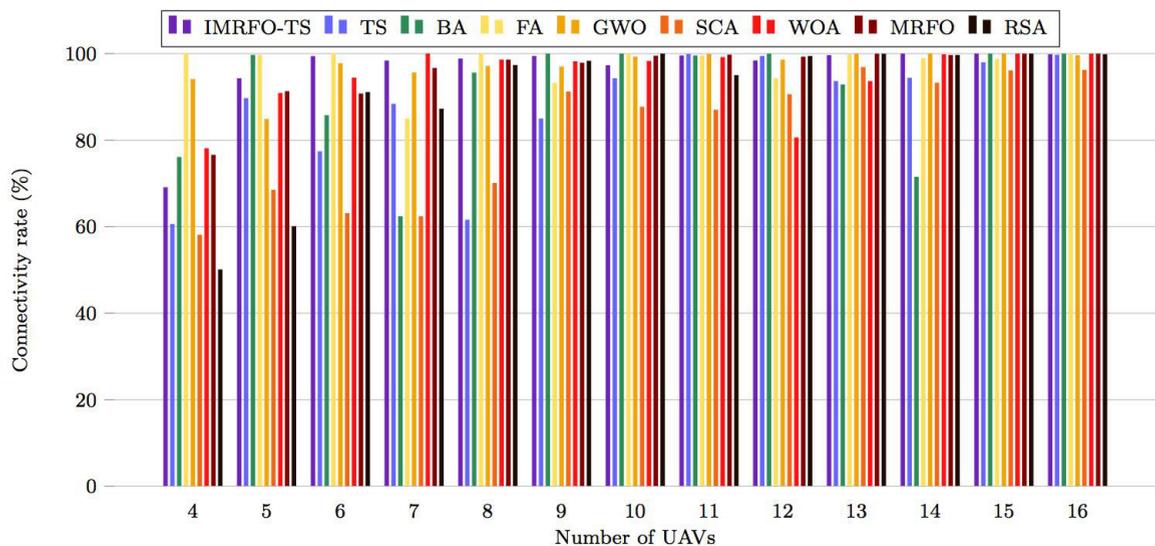


FIGURE 25. The average connectivity rate in the third case.

Table 12 displays the Friedman test results provided by the algorithms, performed in the last case of users. As expected, the optimal descriptive statistic results are provided by the IMRFO-TS algorithm. These results demonstrated the efficiency of our proposed approach in dealing with high number of users, which reflects the high complexity problem. Moreover, with the optimal standard deviation results, we can conclude that the IMRFO-TS is stable along the scenario. This characteristic proved that, our approach is likely getting closer to the global optimal solution. The highest mean rank in this case is also provided by our approach. Both MRFO and GWO offer high mean rank. However, the mean rank gap is important and clearly noticed. Regarding

the significance value, we can notice that it is less than 0.1% also in this case. Therefore, the null hypothesis is rejected.

The whisker plot of the fitness results provided by the algorithms in the last case is presented in Figure 27. As expected, the median fitness varies from an algorithm to another. Therefore, the algorithms statistics are not identical, as proved by the significance value. The distribution of the IMRFOTS, BA, SCA, WOA, MRFO, RSA algorithms is negatively asymmetric. As for the rest, their distribution is positively skewed. None of the algorithms present a normal distribution in this case. Regarding the outliers, IMRFO-TS, TS, GWO, and original MRFO have. Table 13 represents the

TABLE 12. The Friedman tests of the fitness cost in the fourth case.

	Results	IMRFO-TS	TS	BA	FA	GWO	SCA	WOA	MRFO	RSA	
Descriptive statistics	N	13	13	13	13	13	13	13	13	13	
	Minimum	-1.05	-4.7	-4.79	-3.22	-1.32	-3.84	-3.32	-1.39	-5.18	
	Maximum	0.3	-1.78	-1.8	-1.32	0.17	-1.71	-1.57	0.16	-1.65	
	Mean	-0.6	-2.84	-2.87	-2.08	-0.91	-2.56	-2.09	-0.86	-2.96	
	STD	0.46	0.95	0.96	0.61	0.46	0.77	0.52	0.42	1.14	
Rank	Mean rank	8.888	2.31	1.92	5.31	7.27	3.85	5.46	7.62	2	
Statistic tests	N						13				
	Chi-square						96.64				
	Df						8				
	Asymp. sig						< 0.001				

TABLE 13. The post hoc test of the mean fitness of each algorithm in the fourth case.

Multiple Comparison						
Dependent variable: Fitness						
Algorithm (I)	Algorithm (J)	Mean difference (I-J)	SE	Sig	95 % confidence interval	
					Lower bound	Upper bound
IMRFO-TS	TS	2.23	0.33	< 0.001	1.59	2.88
	BA	2.26	0.33	< 0.001	1.62	2.91
	FA	1.47	0.33	< 0.001	0.83	2.12
	GWO	0.31	0.33	0.35	-0.34	0.95
	SCA	1.96	0.33	< 0.001	1.31	2.6
	WOA	1.19	0.33	< 0.001	0.54	1.83
	MRFO	0.25	0.33	0.44	-0.39	0.9
	RSA	2.35	0.33	< 0.001	1.71	3

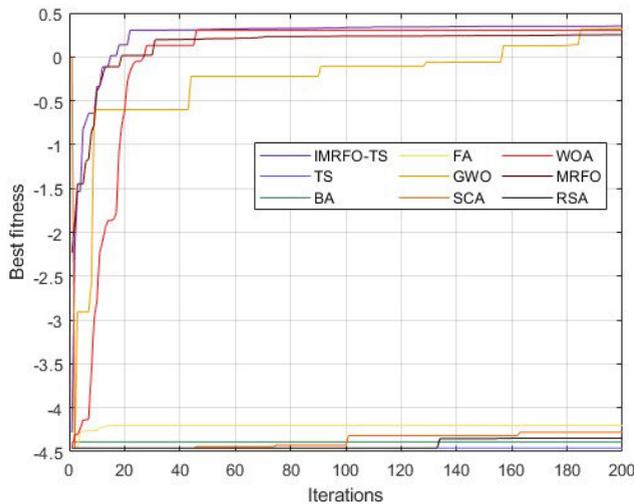


FIGURE 26. Convergence curve in the case of 140 users and 4 UAVs.

post hoc results of the fitness provided by the algorithms in the last case of users. According to the table, we can notice that the significance value is not the same for all the algorithms. Thus, the statistical significance exists for some algorithms. Except for GWO and MRFO algorithms, the significance value is 0.1 % which is less than 5 %. Therefore, the hypothesis that no statistical significance exists is rejected.

Figure 28 depicts the average coverage rate provided by different algorithms under various number of UAVs. As it can be seen, the use of 6 UAVs is the most optimal case of IMRFO-TS regarding the coverage metric. In this case, IMRFO-TS covers approximately 96% of users. Up to

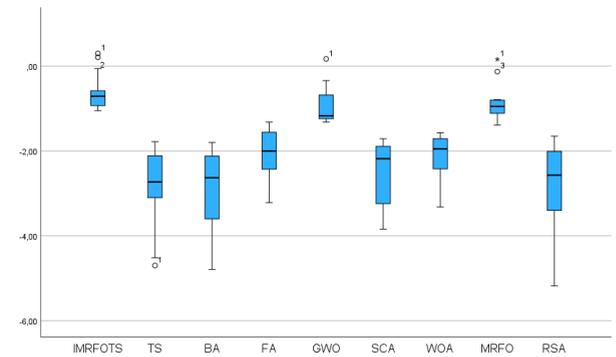


FIGURE 27. The whisker plot of the fitness results obtained in the fourth case.

11 UAVs, IMRFO-TS ensures coverage quality superior to 85% users. GWO and MRFO algorithms offer good coverage quality up to 99 for $U = 4$. GWO and MRFO can cover more than 85% of users for $U \leq 8$ and $U \leq 9$, respectively. WOA achieves coverage rate of 87.27% in case of 4 UAVs. In other cases, it provides less than 85% of users coverage. The other algorithms present a weak coverage rate that is not acceptable in real-life applications.

Figure 29 shows the average UAVs connectivity rate. As noticed, the proposed IMRFO-TS provides an optimal UAVs connectivity results. In most cases, IMRFO-TS offers more than 98% of UAVs connectivity and achieves the 100% of connectivity for $U = 16$. GWO and MRFO also give good connectivity results but not good as IMRFO-TS results. The connectivity quality given by BA, FA, and RSA algorithms if optimal starting from $U = 7$. As for WOA and SCA

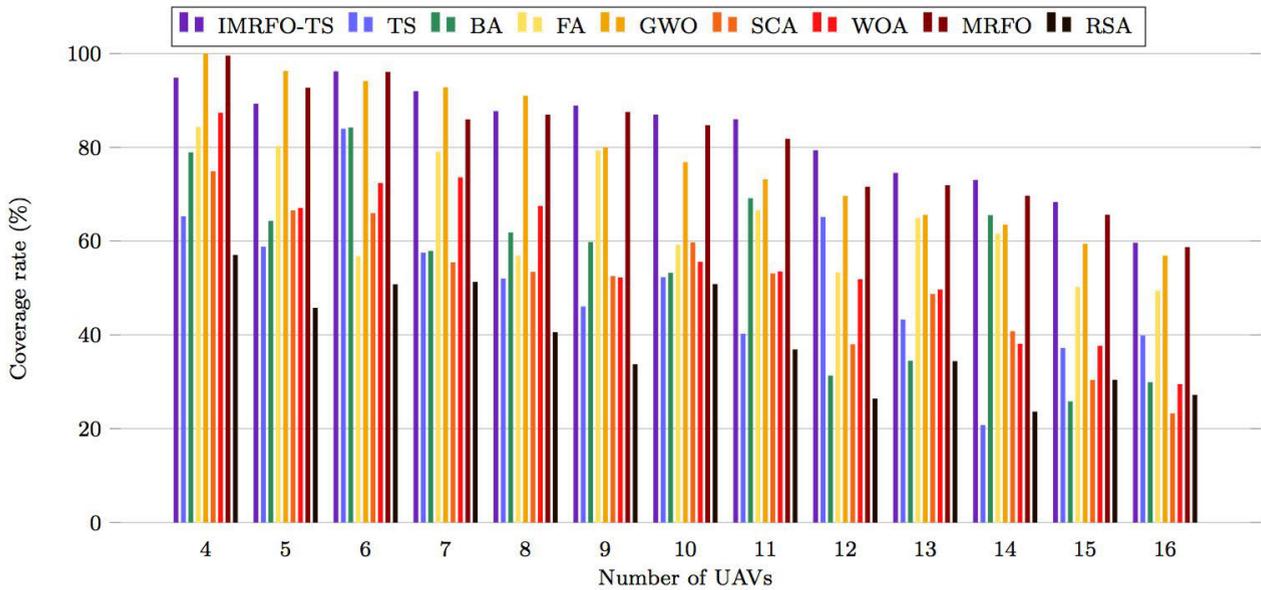


FIGURE 28. The average coverage rate in the fourth case.

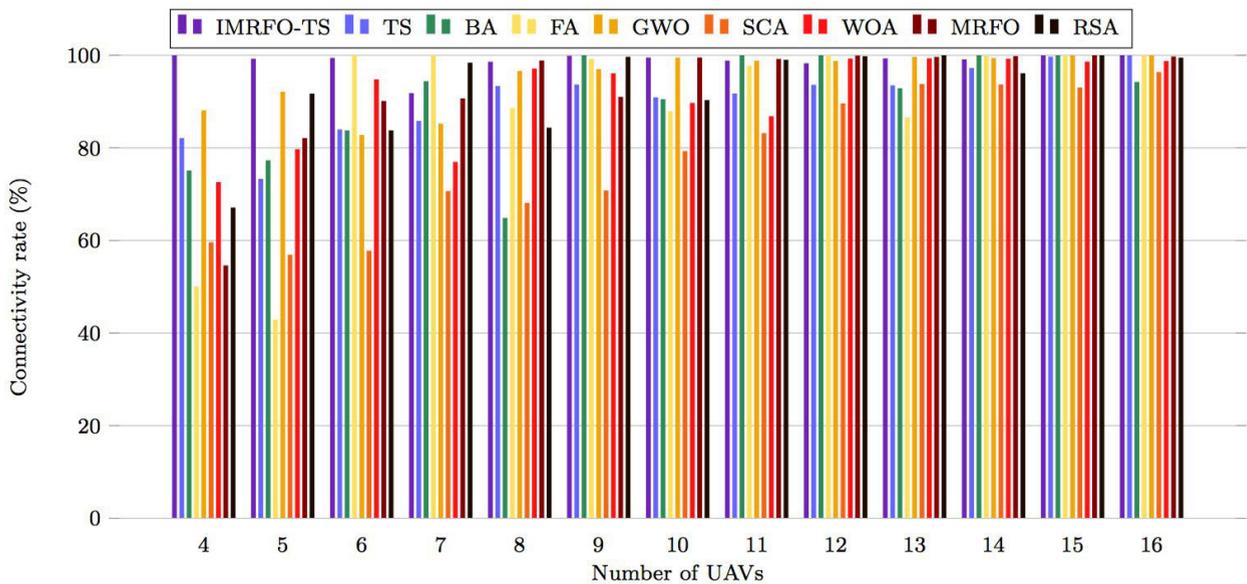


FIGURE 29. The average connectivity rate in the fourth case.

algorithms, they require more than 8 and 12 UAVs, respectively to offer good connectivity quality.

Figure 22 illustrates the average total energy consumed by different UAVs serving 140 users. As we can see, the proposed approach provide the optimal and smallest value of 12.87% in case of $U = 16$. In other cases, the energy rate is remarkable and around 38%, as a result of the coverage quality offered. In this scenario, IMRFO-TS tend to compromise between the coverage rate and energy consumption which are both related to UAVs' heights. It results can be acceptable since IMRFO-TS is able to balance between the two objectives. For $U \geq 7$, results given by GWO

and MRFO are closer to IMRFO-TS results and acceptable. Results given by BA can be acceptable only in case of $U = \{12, 13\}$. SCA and WOA offer optimal results regarding the energy consumption for $U \geq 7$ and $U \geq 10$. FA algorithm presents the worst results that cannot be acceptable.

Figure 23 depicts the average load distribution given by the algorithms under various number of UAVs. From results, it can be seen that the proposed IMRFO-TS provides the smallest load value of 0.02 for $U = 7$. The second best load (2.15) value is given by GWO in case of $U = 7$. By varying the number of UAVs, we can notice that the load distribution

TABLE 14. Results obtained from different algorithms in the fourth case.

Results	IMRFO-TS	TS	BA	FA	GWO	SCA	WOA	MRFO	RSA	
Fitness	0.30	-4.70	-4.23	-2.33	0.17	-3.77	-1.57	0.16	-5.18	U = 4
Coverage	94.76	65.18	78.83	84.26	99.97	74.79	87.27	99.46	56.93	
Connectivity	99.97	82	75	50	88	59.5	72.5	54.5	67	
Energy	49.76	56.35	50.31	57.85	63.19	54.1	57.64	48.66	46.68	
Load	0.02	19.71	2.23	10.08	2.27	1.99	2.16	4.31	21.49	
Fitness	0.21	-4.52	-4.79	-3	-0.34	-3.84	-3.32	-0.8	-4.86	U = 5
Coverage	89.2	58.7	64.21	80.19	96.2	66.44	66.99	92.6	45.66	
Connectivity	99.13	73.2	77.2	42.8	92	56.8	79.6	82	91.6	
Energy	51.26	56.62	42.91	70.07	57.42	45.34	41.6	42.28	41.85	
Load	0.55	18.84	2.19	12.54	2.15	2.02	2.10	4.33	20.4	
Fitness	-0.06	-3.54	-3.6	-3.22	-0.49	-3.3	-2.68	-0.13	-3.88	U = 6
Coverage	96.1	83.87	84.13	56.69	94.04	65.86	72.29	95.99	50.67	
Connectivity	99.3	83	83.67	100	82.67	57.67	94.67	90	83.67	
Energy	43.52	48.12	62.48	51.84	50.6	43.77	57.22	42.59	44.25	
Load	1.77	15.36	2.22	13.94	2.17	2.02	2.15	4.36	16.4	
Fitness	-0.58	-3.08	-3.69	-2.43	-0.68	-3.24	-2.1	-1.39	-3.33	U = 7
Coverage	91.87	57.44	57.79	78.93	92.7	55.36	73.5	85.86	51.19	
Connectivity	91.71	85.71	94.29	100	85.14	70.57	76.86	90.57	98.29	
Energy	40.03	49.13	57.24	62.07	46.59	36.38	46.9	35.48	35.68	
Load	3.78	13.25	2.15	10.9	2.21	2.02	2.18	4.42	14.46	
Fitness	-0.88	-3.1	-3.23	-2.49	-0.71	-2.9	-2.42	-0.95	-3.4	U = 8
Coverage	87.63	51.91	61.73	56.8	90.9	53.34	67.41	86.86	40.47	
Connectivity	98.5	93.25	64.75	88.5	96.5	68	97	98.75	84.25	
Energy	44.15	41.4	53.13	34.05	43.79	36.28	53.99	31.74	40.29	
Load	4.97	13.44	2.23	11.08	2.27	2.07	2.22	4.48	14.44	
Fitness	-0.61	-2.89	-2.77	-2	-1.32	-2.62	-2.48	-0.8	-2.98	U = 9
Coverage	88.79	45.96	59.69	79.21	79.87	52.43	52.14	87.44	33.63	
Connectivity	99.78	93.56	100	99.11	96.89	70.67	96	90.89	99.56	
Energy	34.35	42.24	35.96	58.69	35.11	33.81	42.36	47.71	27.89	
Load	3.98	12.52	2.26	9.2	2.23	2.20	2.16	4.58	12.98	
Fitness	-0.71	-2.73	-2.63	-2.26	-1.28	-2.17	-2.12	-0.79	-2.57	U = 10
Coverage	86.88	52.21	53.13	59.11	76.74	59.6	55.47	84.61	50.73	
Connectivity	99.4	90.8	90.4	87.8	99.4	79.2	89.6	99.4	90.2	
Energy	38.44	47.74	43.08	58.18	35.21	38.38	32.59	39.59	50.98	
Load	4.33	11.87	2.23	9.93	2.25	2.18	2.24	4.64	11.17	
Fitness	-0.69	-2.33	-2.49	-1.56	-1.24	-2.18	-1.95	-0.9	-2.57	U = 11
Coverage	85.89	40.14	69.04	66.44	73.09	52.99	53.43	81.71	36.8	
Connectivity	98.73	91.64	100	97.64	98.73	83.09	86.73	99.09	98.91	
Energy	40.71	44.57	56.97	49.89	29.47	35.33	42.46	37.41	46	
Load	4.22	10.2	2.3	7.39	2.27	2.18	2.25	4.68	11.18	
Fitness	-0.9	-2.21	-2.3	-1.83	-1.23	-2.11	-1.81	-1.21	-2.28	U = 12
Coverage	79.27	65.04	31.21	53.2	69.55	37.9	51.76	71.49	26.3	
Connectivity	98.17	93.5	100	100	98.67	89.5	99.17	99.83	99.67	
Energy	37.21	46.54	33.11	46.01	27.69	22.69	30.3	27.28	23.21	
Load	5	9.95	2.22	8.38	2.28	2.20	2.26	4.57	10.14	
Fitness	-0.93	-2.11	-2.12	-1.69	-1.24	-1.89	-1.76	-1.1	-2.01	U = 13
Coverage	74.44	43.14	34.37	64.79	65.49	48.63	49.57	71.81	34.29	
Connectivity	99.23	93.38	92.77	86.46	99.54	93.69	99.23	99.54	100	
Energy	36.05	43.16	36.5	62.64	27.91	38.61	37.58	38.08	28.67	
Load	5.1	9.42	2.29	7.75	2.33	2.27	2.29	4.67	9.1	
Fitness	-0.95	-2.05	-1.89	-1.32	-1.19	-1.78	-1.71	-1.11	-1.99	U = 14
Coverage	72.94	20.66	65.44	61.47	63.41	40.67	38	69.57	23.53	
Connectivity	99	97.14	100	99.86	99.29	93.57	99.14	99.71	96	
Energy	36.48	41.32	40.43	54.34	28.49	30.9	19.36	28.91	24.8	
Load	5.15	8.96	2.38	6.35	2.38	2.32	2.27	4.77	8.9	
Fitness	-1	-1.82	-1.8	-1.45	-1.17	-1.79	-1.65	-1	-1.76	U = 15
Coverage	68.23	37.11	25.71	50.14	59.34	30.26	37.56	65.51	30.31	
Connectivity	99.86	99.6	100	99.87	99.87	92.93	98.53	100	100	
Energy	37.54	43.5	40.13	48.77	23.95	22.14	31.2	27.16	25.69	
Load	5.44	8.19	2.33	6.83	2.40	2.31	2.36	4.82	8.08	
Fitness	-1.05	-1.78	-1.84	-1.42	-1.12	-1.71	-1.59	-1.12	-1.65	U = 16
Coverage	59.54	39.74	29.81	49.31	56.79	23.14	29.4	58.61	27.1	
Connectivity	100	99.87	94.12	100	100	96.25	98.62	99.62	99.37	
Energy	12.87	36.81	41.57	53.09	24.4	15.31	20.98	29.6	20.44	
Load	5.54	8.14	2.35	6.63	2.44	2.33	2.35	4.89	7.67	

obtained by IMRFO-TS algorithm is decreasing. It can be explained by the fact that the number of covered users is decreasing due to the impact of energy metric. IMRFO-TS results are considered optimal since the gap of load distribution between UAVs does not exceed 3%. In this scenario, load results of BA, GWO, SCA, and WOA algorithms are optimal and less than the one given by IMRFO-TS. But, these values are small only because the coverage rate is reduced. Regarding TS, FA, and RSA results, load values are remarkable and not optimal.

E. COMPARISON BETWEEN DIFFERENT CASES

Comparing tables 5, 15, 16, and 14, we can notice that varying the number of users effects on the optimization results provided by the algorithms. By increasing the number of users, the performance of most algorithms, except IMRFO-TS is decreasing. It can be explained by the fact that the complexity of the problem is increasing with the number of users which make it hard for them to maintain the same solution quality. In the most complex scenario (G = 16), IMRFO-TS provides its best fitness value com-

TABLE 15. Results obtained from different algorithms in the second case.

Results	IMRFO-TS	TS	BA	FA	GWO	SCA	WOA	MRFO	RSA	
<i>Fitness</i>	0.23	-1.88	-1.88	-1.59	0.17	-1.02	-0.07	-0.38	-1.75	<i>U</i> = 4
<i>Coverage</i>	98.8	65.63	63.6	74.3	98.2	82.3	93.8	90.4	68.73	
<i>Connectivity</i>	74.5	75	77.5	100	85.5	69.5	73	80.5	50	
<i>Energy</i>	50.84	52.44	40.62	56.62	57.80	61.81	62.65	44.04	38.62	
<i>Load</i>	0.3	8.44	8.53	7.52	0.56	4.97	1.31	2.8	7.78	
<i>Fitness</i>	0.06	-1.71	-1.48	-1.05	0.2	-1.42	-1.16	-0.42	-1.99	<i>U</i> = 5
<i>Coverage</i>	95.65	58.37	61.2	71.73	97.4	68.2	72.5	88.2	56.13	
<i>Connectivity</i>	84.4	62	100	99.6	98.4	73.2	94	78.8	77.2	
<i>Energy</i>	41.89	49.36	38.28	62.02	51.83	44.18	56.35	38.61	43.47	
<i>Load</i>	1.14	7.54	7.17	5.3	0.65	6.67	5.73	2.95	8.85	
<i>Fitness</i>	0	-1.58	-1.46	-1.21	0.11	-1.11	-0.86	0.2	-1.39	<i>U</i> = 6
<i>Coverage</i>	91.84	59.1	57.5	69.47	96.6	72.8	69	96	62.4	
<i>Connectivity</i>	98	73.33	76	92.67	99.67	63	86	99.67	98.67	
<i>Energy</i>	32.04	49.33	45.58	65.74	51.73	44.96	40.71	40.84	44.34	
<i>Load</i>	1.94	7.16	6.73	5.82	1	5.34	4.6	0.75	6.71	
<i>Fitness</i>	-0.11	-1.37	-1.39	-1.03	-0.34	-1.21	-1.22	-0.12	-1.31	<i>U</i> = 7
<i>Coverage</i>	91.87	51.5	39.1	61.07	86.4	59.7	53.8	91.4	49.67	
<i>Connectivity</i>	97.43	95.43	100	99.71	98.86	82.29	89.14	98	68.86	
<i>Energy</i>	36.87	42.38	42.03	49.39	37.45	41.62	39.46	33.42	42.69	
<i>Load</i>	1.99	6.53	6.53	5.24	2.85	5.84	5.92	2.04	6.01	
<i>Fitness</i>	-0.05	-1.13	-1.33	-0.87	-0.23	-1.03	-1.05	0	-1.29	<i>U</i> = 8
<i>Coverage</i>	92.9	62.3	41.9	62.67	88.9	55.5	51.8	98.5	35.7	
<i>Connectivity</i>	99.75	87.75	71	100	98.5	86.5	98.75	94.07	96	
<i>Energy</i>	33.68	45.69	41.9	47.17	39.27	33.92	38.35	34.33	37.32	
<i>Load</i>	1.80	5.58	4.98	4.65	2.42	5.22	5.33	1.57	6.1	
<i>Fitness</i>	-0.16	-1.1	-1.20	-0.69	-0.35	-0.98	-0.68	-0.27	-1.02	<i>U</i> = 9
<i>Coverage</i>	85.47	62.07	48.2	71.57	80.2	46.4	66.4	83	35.87	
<i>Connectivity</i>	100	83.33	100	99.78	98.22	92.89	99.78	96.89	100	
<i>Energy</i>	31.29	40.82	52.05	52.51	34.15	26.23	49.08	26.33	27.15	
<i>Load</i>	2.21	5.43	5.76	3.95	2.83	5.05	3.89	2.6	5.19	
<i>Fitness</i>	-0.23	-0.97	-1.04	-0.7	-0.5	-0.85	-0.9	-0.26	-1.02	<i>U</i> = 10
<i>Coverage</i>	83.03	57.87	51.7	62.63	72.4	49.2	48.27	80.3	32.07	
<i>Connectivity</i>	100	90.6	99.6	98.4	98.8	93.8	99.8	100	99	
<i>Energy</i>	34.48	46.01	43.27	42.17	28.98	27.21	37.52	31.88	29.6	
<i>Load</i>	2.43	4.91	5.23	3.98	3.4	4.54	4.71	2.52	5.08	
<i>Fitness</i>	-0.23	-0.89	-0.91	-0.71	-0.33	-0.75	-0.72	-0.29	-0.76	<i>U</i> = 11
<i>Coverage</i>	79.87	69.53	34.8	56.47	76.9	50	50.1	78.7	49.27	
<i>Connectivity</i>	99.45	100	100	97.45	100	95.64	99.45	99.45	98.73	
<i>Energy</i>	31.11	39.46	40.46	48.23	29.05	28.6	34.14	28.8	33.65	
<i>Load</i>	2.42	4.85	4.59	3.92	2.79	4.17	4.05	2.93	4.19	
<i>Fitness</i>	-0.27	-0.64	-0.89	-0.62	-0.32	-0.72	-0.68	-0.32	-0.74	<i>U</i> = 12
<i>Coverage</i>	77.8	68.33	31.4	60.5	74.9	41.8	45.27	72.2	36.27	
<i>Connectivity</i>	99.83	91.67	92.33	100	99.33	97.17	100	99.67	99.5	
<i>Energy</i>	34.87	38.03	40.87	51.15	23.92	21.05	31.32	23.13	20.85	
<i>Load</i>	2.52	3.8	4.4	3.59	2.78	4.07	3.86	2.76	4.09	
<i>Fitness</i>	-0.23	-0.74	-0.76	-0.46	-0.36	-0.73	-0.6	-0.36	-0.76	<i>U</i> = 13
<i>Coverage</i>	75.07	35	60.6	67.77	65.4	31	50.9	64.5	20.13	
<i>Connectivity</i>	99.84	100	100	92.15	99.54	96.92	99.85	100	99.85	
<i>Energy</i>	25.81	36.19	37.74	44	25.17	22.11	39.91	25.87	15.66	
<i>Load</i>	2.41	3.92	4.28	3.01	2.83	4	3.49	2.81	4.1	
<i>Fitness</i>	-0.29	-0.75	-0.74	-0.51	-0.4	-0.56	-0.63	-0.3	-0.68	<i>U</i> = 14
<i>Coverage</i>	71.36	52.9	44.4	59.9	60.7	54.6	36.2	66.7	39.8	
<i>Connectivity</i>	100	98.71	99	100	99.57	97.29	99.71	99.86	99.71	
<i>Energy</i>	32.79	44.04	31.36	52.45	23.43	33.76	27.76	25.77	35.55	
<i>Load</i>	2.55	4.08	4.08	3.11	2.95	3.42	3.58	2.63	3.77	
<i>Fitness</i>	-0.17	-0.65	-0.68	-0.49	-0.29	-0.63	-0.58	-0.35	-0.59	<i>U</i> = 15
<i>Coverage</i>	73.67	43.43	25.6	55.87	69.2	27.1	33.3	62.9	33.47	
<i>Connectivity</i>	99.87	99.87	99.87	100	99.87	98.4	99.73	99.87	99.37	
<i>Energy</i>	22.53	43.76	36.44	49.13	25.18	17.86	25.39	28.67	33.97	
<i>Load</i>	2.19	3.6	3.62	3.04	2.59	3.6	3.41	2.74	3.33	
<i>Fitness</i>	-0.25	-0.63	-0.68	-0.36	-0.29	-0.49	-0.46	-0.35	-0.56	<i>U</i> = 16
<i>Coverage</i>	61.87	41.47	68.7	61.4	59.2	47	63.6	48.7	31.2	
<i>Connectivity</i>	100	99.37	99.75	100	99.87	97.18	99.87	99.75	99.75	
<i>Energy</i>	19.65	40.74	47.15	45.38	19.97	38.48	47.31	14.35	16.04	
<i>Load</i>	2.46	3.53	3.92	2.59	2.53	3.02	3	2.74	3.39	

paring to the other scenarios. The reason behinds is that the improvement and hybridization strategies of IMRFO-TS algorithm are efficient and make difference compared to other algorithms.

The coverage quality is also impacted by the number of users. Comparing the optimal coverage value given by the

algorithms using different users, we can conclude that the coverage rate is increasing with the number of users. It can be justified by the fact that more users are present within the coverage area. From another side, keeping the same coverage quality, additional UAVs are required. In fact, the coverage rate is calculated based on the distance between

TABLE 16. Results obtained from different algorithms in the third case.

Results	IMRFO-TS	TS	BA	FA	GWO	SCA	WOA	MRFO	RSA	
<i>Fitness</i>	0.19	-3.04	-3.6	-2.85	-0.08	-1.57	-2.74	-0.18	-2.97	<i>U</i> = 4
<i>Coverage</i>	99.1	70.58	55.16	76.14	96.7	83.18	76.04	92.22	73.58	
<i>Connectivity</i>	69	60.5	76	100	94	58	78	76.5	50	
<i>Energy</i>	44.29	42.23	56.23	64.64	58.47	57.34	48.87	51.34	54.91	
<i>Load</i>	0.45	13.07	1.8	12.51	1.67	1.62	1.74	3.6	12.56	
<i>Fitness</i>	-0.23	-3.05	-3.58	-2	-0.71	-2.75	-1.34	-0.34	-3	<i>U</i> = 5
<i>Coverage</i>	78.4	71.9	37.66	77.52	91.12	60.84	83.78	99.82	59	
<i>Connectivity</i>	94.2	89.6	99.6	99.6	84.8	68.4	90.8	91.2	60	
<i>Energy</i>	39.82	54.21	42.46	57.1	48.28	44.90	67.52	47.10	43.81	
<i>Load</i>	2.27	13.27	1.74	9.22	1.81	1.64	1.79	3.61	12.74	
<i>Fitness</i>	0.11	-2.67	-2.72	-1.38	-0.07	-2.65	-1.92	-0.24	-2.55	<i>U</i> = 6
<i>Coverage</i>	98.16	63.64	50.58	85.86	96.94	53.28	68.1	93.54	53.54	
<i>Connectivity</i>	99.3	77.3	85.67	100	97.67	63	94.33	90.67	91	
<i>Energy</i>	37.389	50.46	30.18	54.07	49.52	38.97	51.08	41.55	44.18	
<i>Load</i>	1.17	11.64	1.77	6.83	1.85	1.65	1.77	3.62	11.21	
<i>Fitness</i>	-0.35	-2.29	-2.36	-1.98	-0.54	-2.04	-1.63	-0.37	-1.74	<i>U</i> = 7
<i>Coverage</i>	91.62	65.64	75.46	53.62	90.24	62.76	72.54	96.57	79.88	
<i>Connectivity</i>	98.28	88.29	62.29	84.86	95.53	62.29	100	90.32	87.14	
<i>Energy</i>	35.67	56.48	53.45	41.05	45.13	38.74	41.6	43.78	44.78	
<i>Load</i>	2.95	10.12	1.88	8.88	1.84	1.72	1.79	2.93	8.16	
<i>Fitness</i>	-0.25	-2.25	-2.31	-1.64	-0.57	-2.06	-1.96	-0.35	-2.45	<i>U</i> = 8
<i>Coverage</i>	92.28	48.48	50.4	64.84	88.88	49.58	53.1	88.36	27.66	
<i>Connectivity</i>	98.75	61.5	95.5	100	97	70	98.5	98.5	97.25	
<i>Energy</i>	30.9	40.27	31.98	52.04	40.97	32.16	47.17	36.73	36.53	
<i>Load</i>	2.64	9.7	1.84	7.68	1.89	1.73	1.79	3.66	10.68	
<i>Fitness</i>	-0.36	-2.03	-1.85	-1.66	-0.72	-1.83	-1.62	-0.4	-2.12	<i>U</i> = 9
<i>Coverage</i>	88.4	47.56	43.66	58.56	83.14	57.28	58.9	87.48	45.28	
<i>Connectivity</i>	99.33	84.89	100	93.11	96.89	91.11	98.11	97.78	98.22	
<i>Energy</i>	40.43	42.29	39.96	47.64	35.36	46.23	44.65	35.93	44.12	
<i>Load</i>	2.91	9.02	1.81	7.67	1.88	1.77	1.79	3.65	9.46	
<i>Fitness</i>	-0.65	-1.84	-1.62	-1.3	-0.66	-1.73	-1.5	-0.78	-1.69	<i>U</i> = 10
<i>Coverage</i>	79.66	46.38	63	64.66	81.4	40.36	53.84	75.88	41.32	
<i>Connectivity</i>	97.2	94.2	100	100	99.2	87.6	98.2	99.4	100	
<i>Energy</i>	30.45	39.94	33.88	54.47	35.06	27.23	50.85	28.2	35.29	
<i>Load</i>	4.10	8.34	1.88	6.29	1.9	1.77	1.83	3.67	7.81	
<i>Fitness</i>	-0.43	-1.62	-1.71	-1.2	-0.8	-1.52	-1.41	-0.54	-1.61	<i>U</i> = 11
<i>Coverage</i>	82.76	49.2	31.52	55.34	75.08	50.04	49.36	82.04	43.58	
<i>Connectivity</i>	99.45	99.82	99.45	99.45	99.82	86.91	99.09	99.64	94.91	
<i>Energy</i>	33.24	46.77	35.79	32.12	28.44	31.8	37.53	33.81	38.59	
<i>Load</i>	3.24	7.52	1.84	6.02	1.93	1.85	1.84	3.79	7.46	
<i>Fitness</i>	-0.74	-1.6	-1.5	-1.45	-0.84	-1.4	-1.23	-0.81	-1.62	<i>U</i> = 12
<i>Coverage</i>	73.80	55.9	43.18	44.48	68.56	42.56	53.74	71.96	18.98	
<i>Connectivity</i>	98.3	99.33	100	94.17	98.5	90.5	80.5	99.17	99.33	
<i>Energy</i>	35.74	42.25	38.89	49.17	28	28.59	39.77	34.02	15.83	
<i>Load</i>	4.36	7.51	1.87	6.7	1.94	1.88	1.89	3.84	7.49	
<i>Fitness</i>	-0.53	-1.41	-1.57	-1.12	-0.8	-1.41	-1.16	-0.48	-1.38	<i>U</i> = 13
<i>Coverage</i>	77.48	33.16	23.68	48.62	54	28.6	48.9	80.34	38.56	
<i>Connectivity</i>	99.54	93.54	92.77	99.69	99.85	96.77	93.54	99.85	99.85	
<i>Energy</i>	38.39	39.04	39.19	45.99	24.86	20.08	33.42	31.97	25.47	
<i>Load</i>	3.54	6.52	1.86	5.49	1.96	1.88	1.89	3.89	6.65	
<i>Fitness</i>	-0.65	-1.35	-1.27	-1.07	-0.78	-1.18	-1.02	-0.66	-1.27	<i>U</i> = 14
<i>Coverage</i>	69	62.6	40.6	53.2	62.74	47.38	53.7	69.74	35.68	
<i>Connectivity</i>	100	94.29	71.43	98.86	100	93.14	99.71	99.57	99.57	
<i>Energy</i>	29.82	47.77	37.31	48.52	23.55	32.24	31.38	25.8	27.25	
<i>Load</i>	3.9	6.48	1.92	5.3	1.98	1.95	1.93	3.93	6.14	
<i>Fitness</i>	-0.62	-1.31	-1.29	-0.96	-0.7	-1.21	-0.96	-0.62	-1.18	<i>U</i> = 15
<i>Coverage</i>	68.28	45.98	23.9	59.68	62.56	28.16	54	67.2	29.26	
<i>Connectivity</i>	100	97.87	100	98.67	100	96	100	99.87	100	
<i>Energy</i>	37.22	46.28	41.68	50.15	27.11	18.66	40.11	25.48	23.69	
<i>Load</i>	3.87	6.23	1.91	4.91	2.02	1.95	1.98	3.97	5.79	
<i>Fitness</i>	-0.64	-1.22	-1.14	-0.9	-0.65	-1.09	-0.91	-0.64	-1.16	<i>U</i> = 16
<i>Coverage</i>	64.54	40.46	55.58	57.08	65.48	28.14	59.82	61.62	19.1	
<i>Connectivity</i>	99.75	99.62	100	100	99.5	96.12	99.87	99.87	99.75	
<i>Energy</i>	25.86	43.28	48.51	45.51	25.13	16.98	33.13	24.64	17.85	
<i>Load</i>	3.97	5.83	2.01	4.71	2.05	1.98	2.09	4.02	5.66	

UAVs and users. Additional users involve additional distances to take into consideration. Since the users are randomly distributed to the area and the coverage radius is limited, the probability of the calculated distances to exceed the coverage radius is increasing with the number of users. From one case of users to another, IMRFO-TS requires only 1 more UAV that is considered optimal in terms of costs.

As for the connectivity rate, the number of users is not really affecting on the UAVs connectivity. Most algorithms were able to keep the same connectivity quality for different number of users. The connectivity metric is impacted only by the number of UAVs since the network links are established only between UAVs.

Regarding the energy metric, it can be noticed that it changes according to the number of users. The energy

consumption is related to users via the coverage metric. As mentioned before, coverage rate is increasing with the users which implies the rise of UAVs' height. As results, the energy efficiency is decreasing. From this fact, we can conclude that the energy and coverage metrics are trade-offs. In most user cases, the proposed IMRFO-TS approach offers a good balance between them. MRFO and GWO also present good results. SCA, and RSA algorithms tend to optimize the energy parameter over the coverage. The other algorithms provide high energy results.

Load distribution metric is impacted directly by the number of covered users. By adding more users, the coverage quality is improved. As results, the UAVs are able to share more users which reduces the gap of the load distribution. Considering the optimal load distribution results, IMRFO-TS outperforms all meta-heuristics by given the smallest value.

VII. CONCLUSION

In this paper, we have introduced a hybrid solution (IMRFO-TS) based on the combination of improved MRFO with TS algorithm for solving the UAV deployment issue in a smart city. A fitness function is modeled considering the user coverage, UAV connectivity, energy consumption, and load distribution parameters. The effectiveness of the proposed IMRFO-TS algorithm is assessed using 52 benchmarks in comparison with the original MRFO and eight competitor optimization meta-heuristics such as TS, BA, FA, GWO, SCA, WOA, and RSA. Evaluation of the simulation results demonstrated the performance and efficiency of the proposed IMRFO-TS algorithm by finding optimal locations of UAVs.

As for future works, several research directions can be considered. First, the UAV placement problem can be modeled as a multi-objective problem based on the Pareto front. Second, to deal with real-time applications, UAVs and users mobility can be considered. Finally, resource allocation for users is an important point that can be addressed.

APPENDIX

See Tables 15 and 16.

REFERENCES

- [1] S. Sabino and A. Grilo, "Topology control of unmanned aerial vehicle (UAV) mesh networks: A multi-objective evolutionary algorithm approach," in *Proc. 4th ACM Workshop Micro Aerial Vehicle Netw., Syst., Appl.*, Jun. 2018, pp. 45–50.
- [2] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103300.
- [3] E. H. Houssein, G. N. Zaki, A. A. Z. Diab, and E. M. G. Younis, "An efficient manta ray foraging optimization algorithm for parameter extraction of three-diode photovoltaic model," *Comput. Electr. Eng.*, vol. 94, Sep. 2021, Art. no. 107304.
- [4] H. Xu, H. Song, C. Xu, X. Wu, and N. Yousefi, "Exergy analysis and optimization of a HT-PEMFC using developed manta ray foraging optimization algorithm," *Int. J. Hydrogen Energy*, vol. 45, no. 55, pp. 30932–30941, Nov. 2020.
- [5] E. H. Houssein, I. E. Ibrahim, N. Neggaz, M. Hassaballah, and Y. M. Wazery, "An efficient ECG arrhythmia classification method based on manta ray foraging optimization," *Expert Syst. Appl.*, vol. 181, Nov. 2021, Art. no. 115131.
- [6] E. H. Houssein, M. M. Emam, and A. A. Ali, "Improved manta ray foraging optimization for multi-level thresholding using COVID-19 CT images," *Neural Comput. Appl.*, vol. 33, no. 24, pp. 16899–16919, Dec. 2021.
- [7] G. Hu, M. Li, X. Wang, G. Wei, and C.-T. Chang, "An enhanced manta ray foraging optimization algorithm for shape optimization of complex CCG-ball curves," *Knowl.-Based Syst.*, vol. 240, Mar. 2022, Art. no. 108071.
- [8] U. C. Ben, A. E. Akpan, E. O. Enyinyi, and E. Awak, "Novel technique for the interpretation of gravity anomalies over geologic structures with idealized geometries using the manta ray foraging optimization," *J. Asian Earth Sci.*, X, vol. 6, Dec. 2021, Art. no. 100070.
- [9] K. K. Ghosh, R. Guha, S. K. Bera, N. Kumar, and R. Sarkar, "S-shaped versus V-shaped transfer functions for binary manta ray foraging optimization in feature selection problem," *Neural Comput. Appl.*, vol. 33, no. 17, pp. 11027–11041, Jan. 2021.
- [10] B. Sheng, T. Pan, Y. Luo, and K. Jermsittiparsert, "System identification of the PEMFCs based on balanced manta-ray foraging optimization algorithm," *Energy Rep.*, vol. 6, pp. 2887–2896, Nov. 2020.
- [11] A. Fathy, H. Rezk, and D. Yousofi, "A robust global MPPT to mitigate partial shading of triple-junction solar cell-based system using manta ray foraging optimization algorithm," *Sol. Energy*, vol. 207, pp. 305–316, Sep. 2020.
- [12] R. M. Rizk-Allah, M. I. Zineldin, A. A. A. Mousa, S. Abdel-Khalek, M. S. Mohamed, and V. Šnášel, "On a novel hybrid manta ray foraging optimizer and its application on parameters estimation of lithium-ion battery," *Int. J. Comput. Intell. Syst.*, vol. 15, no. 1, pp. 1–22, Aug. 2022.
- [13] Y. Dong, F. Liu, X. Lu, Y. Lou, Y. Ma, and N. Eghbalian, "Multi-objective economic environmental energy management microgrid using hybrid energy storage implementing and developed manta ray foraging optimization algorithm," *Electr. Power Syst. Res.*, vol. 211, Oct. 2022, Art. no. 108181.
- [14] N. Firouz, M. Masdari, A. B. Sangar, and K. Majidzadeh, "A novel controller placement algorithm based on network partitioning concept and a hybrid discrete optimization algorithm for multi-controller software-defined networks," *Cluster Comput.*, vol. 24, no. 3, pp. 2511–2544, Sep. 2021.
- [15] A. A. A. Razak, A. N. K. Nasir, N. M. A. Ghani, N. M. Rizal, M. F. M. Jusof, and I. H. Muhamad, "Spiral-based manta ray foraging optimization to optimize PID control of a flexible manipulator," in *Proc. Emerg. Technol. Comput., Commun. Electron. (ETCCE)*, Dec. 2020, pp. 1–6.
- [16] M. F. M. Jusof, S. Mohammad, A. A. A. Razak, N. M. Rizal, A. N. K. Nasir, and M. A. Ahmad, "Hybrid manta ray foraging—Particle swarm algorithm for PD control optimization of an inverted pendulum," in *Recent Trends in Mechatronics Towards Industry 4.0*. Cham, Switzerland: Springer, 2022, pp. 1–13.
- [17] M. Micev, M. Calasan, Z. M. Ali, H. M. Hasanien, and S. H. E. A. Aleem, "Optimal design of automatic voltage regulation controller using hybrid simulated annealing—Manta ray foraging optimization algorithm," *Ain Shams Eng. J.*, vol. 12, no. 1, pp. 641–657, Mar. 2021.
- [18] H. Sharma and A. S. Jalal, "An improved attention and hybrid optimization technique for visual question answering," *Neural Process. Lett.*, vol. 54, no. 1, pp. 709–730, Feb. 2022.
- [19] K. Elmaadawy, M. A. Elaziz, A. H. Elsheikh, A. Moawad, B. Liu, and S. Lu, "Utilization of random vector functional link integrated with manta ray foraging optimization for effluent prediction of wastewater treatment plant," *J. Environ. Manage.*, vol. 298, Nov. 2021, Art. no. 113520.
- [20] H. D. Nguyen, Q.-H. Nguyen, Q. V. V. Du, T. H. T. Nguyen, T. G. Nguyen, and Q.-T. Bui, "A novel combination of deep neural network and manta ray foraging optimization for flood susceptibility mapping in Quang Ngai province, Vietnam," *Geocarto Int.*, vol. 2021, pp. 1–25, Sep. 2021.
- [21] I. Attiya, M. A. Elaziz, L. Abualigah, T. N. Nguyen, and A. A. A. El-Latif, "An improved hybrid swarm intelligence for scheduling IoT application tasks in the cloud," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6264–6272, Sep. 2022.
- [22] M. Haris and S. Zubair, "Mantaray modified multi-objective Harris hawk optimization algorithm expedites optimal load balancing in cloud computing," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9696–9709, Nov. 2022.
- [23] H. Wang, D. Huo, and B. Alidaee, "Position unmanned aerial vehicles in the mobile ad hoc network," *J. Intell. Robot. Syst.*, vol. 74, nos. 1–2, pp. 455–464, Apr. 2014.

- [24] I. Strumberger, N. Bacanin, S. Tomic, M. Beko, and M. Tuba, "Static drone placement by elephant herding optimization algorithm," in *Proc. 25th Telecommun. Forum (TELFOR)*, Nov. 2017, pp. 1–4.
- [25] S. U. Rahman, G.-H. Kim, Y.-Z. Cho, and A. Khan, "Positioning of UAVs for throughput maximization in software-defined disaster area UAV communication networks," *J. Commun. Netw.*, vol. 20, no. 5, pp. 452–463, 2018.
- [26] D. G. Reina, H. Tawfik, and S. L. Toral, "Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks," *Ad Hoc Netw.*, vol. 68, pp. 16–32, Jan. 2018.
- [27] F. Al-Turjman, L. P. Lemayian, S. Alturjman, and L. Mostarda, "Enhanced deployment strategy for the 5G drone-BS using artificial intelligence," *IEEE Access*, vol. 7, pp. 75999–76008, 2019.
- [28] E. Chaalal, L. Reynaud, and S. M. Senouci, "A social spider optimisation algorithm for 3D unmanned aerial base stations placement," in *Proc. IFIP Netw. Conf., Netw.*, 2020, pp. 544–548.
- [29] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV-enabled load-balance mobile-edge computing for IoT networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6898–6908, Aug. 2020.
- [30] M. Gupta and S. Varma, "Optimal placement of UAVs of an aerial mesh network in an emergency situation," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 343–358, Jan. 2021.
- [31] M. Gupta and S. Varma, "Metaheuristic-based optimal 3D positioning of UAVs forming aerial mesh network to provide emergency communication services," *IET Commun.*, vol. 15, no. 10, pp. 1297–1314, Jun. 2021.
- [32] A. Sawalmeh, N. S. Othman, G. Liu, A. Khreishah, A. Alenezi, and A. Alanazi, "Power-efficient wireless coverage using minimum number of UAVs," *Sensors*, vol. 22, no. 1, p. 223, Dec. 2021.
- [33] V. Mayor, R. Estepa, and A. Estepa, "QoS-aware multilayer UAV deployment to provide VoWiFi service over 5G networks," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–13, Jan. 2022.
- [34] R. Ozdag, "Multi-metric optimization with a new metaheuristic approach developed for 3D deployment of multiple drone-BSSs," *Peer-Peer Netw. Appl.*, vol. 15, no. 3, pp. 1535–1561, May 2022.
- [35] M. A. Ouamri, M.-E. Oteteanu, G. Barb, and C. Gueguen, "Coverage analysis and efficient placement of drone-BSS in 5G networks," *Eng. Proc.*, vol. 14, no. 1, p. 18, 2022.
- [36] P. Mahajan, A. Kumar, G. S. S. Chalapathi, and R. Buyya, "EFTA: An energy-efficient, fault-tolerant, and area-optimized UAV placement scheme for search operations," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2022, pp. 1–6.
- [37] W. Wei, J. Wang, Z. Fang, J. Chen, Y. Ren, and Y. Dong, "3U: Joint design of UAV-USV-UUV networks for cooperative target hunting," *IEEE Trans. Veh. Technol.*, early access, Nov. 9, 2022.
- [38] Y. Liu, W. Huangfu, H. Zhou, H. Zhang, J. Liu, and K. Long, "Fair and energy-efficient coverage optimization for UAV placement problem in the cellular network," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 4222–4235, Jun. 2022.
- [39] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *Proc. 12th Int. Conf. Adv. Robot.*, 2005, pp. 492–497.
- [40] L. Di Puglia Pugliese, F. Guerriero, D. Zorbas, and T. Razafindralambo, "Modelling the mobile target covering problem using flying drones," *Optim. Lett.*, vol. 10, no. 5, pp. 1021–1052, Jun. 2016.
- [41] F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, Aug. 1990.
- [42] M. Qiu, Z. Fu, R. Eglese, and Q. Tang, "A Tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups," *Comput. Oper. Res.*, vol. 100, pp. 102–116, Dec. 2018.
- [43] X. Chou, L. M. Gambardella, and R. Montemanni, "A Tabu search algorithm for the probabilistic orienteering problem," *Comput. Oper. Res.*, vol. 126, Feb. 2021, Art. no. 105107.
- [44] F. Khafa, C. Sánchez, A. Barolli, and M. Takizawa, "Solving mesh router nodes placement problem in wireless mesh networks by Tabu search algorithm," *J. Comput. Syst. Sci.*, vol. 81, no. 8, pp. 1417–1428, Dec. 2015.
- [45] J. Van Belle, P. Valckenaers, G. Vanden Berghe, and D. Cattrysse, "A Tabu search approach to the truck scheduling problem with multiple docks and time windows," *Comput. Ind. Eng.*, vol. 66, no. 4, pp. 818–826, Dec. 2013.
- [46] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "Grasshopper optimization algorithm: Theory, variants, and applications," *IEEE Access*, vol. 9, pp. 50001–50024, 2021.
- [47] S. L. Tilahun, "Balancing the degree of exploration and exploitation of swarm intelligence using parallel computing," *Int. J. Artif. Intell. Tools*, vol. 28, no. 3, May 2019, Art. no. 1950014.
- [48] X. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, Jul. 2012.
- [49] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [50] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
- [51] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2019.
- [52] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Syst. Appl.*, vol. 191, Apr. 2022, Art. no. 116158.



AMYLIA AIT SAADI received the master's degree from the University of Science and Technology Houari Boumediene, Algeria, in 2019. She is currently pursuing the Ph.D. degree with the LIST Laboratory, University M'Hamed Bougara Boumerdes, Algeria, and the LISV Laboratory, University of Paris-Saclay, France. Her research interests include unmanned aerial vehicles, path planning, and the application of meta-heuristics for optimization problems.



ASSIA SOUKANE is currently the Director of research and responsible for the intelligent communicating systems axis with the ECE Paris School of Engineering. Her research began with a "DEA" followed by a Ph.D. thesis; both were carried out at the DAVID Laboratory, University of Versailles. Her research interest includes the design of multi-source systems in heterogeneous environments.



YASSINE MERAIHI received the Ph.D. degree from the University of M'Hamed Bougara Boumerdes, Algeria, in 2017. He is currently an Associate Professor with the University of M'Hamed Bougara Boumerdes. His research interests include QoS for wireless networks, routing in challenged networks, including WMSNs/VANETs, and the applications of meta-heuristics to optimization problems.



ASMA BENMESSAOUD GABIS received the Ph.D. degree from Ecole Nationale Supérieure d'Informatique, Algiers, Algeria, in 2021. Her research interests include network design and communication, the application of AI, machine learning, and meta-heuristics for multi-objective optimization and performance evaluation.



AMAR RAMDANE-CHERIF received the Ph.D. degree from Pierre and Marie Curie University, Paris, in 1998. Since 2000, he has been a Professor with the University of Paris-Saclay, France. His research interests include software architecture, dynamic architecture, architectural quality attributes, architectural styles, and design patterns.